



电子科技大学

University of Electronic Science and Technology of China

学士学位论文

BACHELOR DISSERTATION

论文题目人体检测及行为分析的研究

学生姓名 _____ 王官皓 _____

学号 _____ 2010021080018 _____

专业 _____ 电子信息工程 _____

学院 _____ 电子工程学院 _____

指导教师 _____ 张翔 _____

指导单位 _____ 电子科技大学 _____

年月日

摘要

人体检测及行为分析是计算机视觉中快速发展的一个领域、在智能汽车、监控系统和高级机器人等方面具有关键性作用。这篇文章的目的是从方法学和具体实现的角度考察目前具有代表性的人体检测及行为分析系统的主要组件和底层模型，包括人体检测方面：基于Haar小波的AdaBoost级联器[1]、HOG/linSVM[2]，以及行为分析方面：STIP/SVM[3]。在行人检测方面采用Daimler提供的泛数据集以及采集于校园环境中的实时视频。在行为分析方面考察采用电影剧本的自动标注方法来处理视频数据集的限制，恢复出的行为样本用于视觉学习。考察的系统对特定任务呈现出的可观的结果。

关键词：人体检测及行为分析， HOG/linSVM， Haar/AdaBoost， STIP， 自动标注

ABSTRACT

ABSTRACT

Human Detection and Action Recognition are rapidly evolving areas in computer vision with key applications in intelligent vehicles, surveillance, and advanced robotics. The objective of this paper is to provide an inspection of several current state-of-the-art human detection and action recognition systems from both methodological and implementation perspectives, including human detection: Haar/AdaBoost[1], HOG/linSVM[2], as well as Action Recognition:STIP/SVM[3]. In human detection, implementations are performed on an extensive data set provided by Daimler as well as videos captured through campus environment. While in action recognition this paper firstly evaluates automatic annotation methods using scripts to handle the limitation of video data set, then uses the retrieved action samples for visual learning and recognition. Systems considered in this paper show promising results in application-specific scenario.

Keywords: Human Detection&Action Recognition, HOG/linSVM, Haar/AdaBoost, STIP, Automatic Annotation

目 录

第1章 引言	1
1.1 背景介绍	1
1.2 研究现状	1
1.2.1 主要部件	2
1.2.1.1 假设生成	2
1.2.1.2 模型匹配	2
1.2.2 特征描述与提取算法	3
1.2.3 学习算法	3
1.2.4 数据集合	4
1.2.5 存在问题	5
1.3 本文研究内容	6
1.4 本文结构	6
第2章 人体检测	7
2.1 代表性算法	7
2.2 特征	7
2.2.1 Haar特征	7
2.2.2 HOG特征	9
2.3 分类器	13
2.3.1 AdaBoost级联器	13
2.3.2 支持向量机(SVM)	16
第3章 行为分析	29
3.1 训练样本构建	29
3.2 STIP特征	31
3.3 分类器	35
第4章 具体实现及效果	36
4.1 运算库	36
4.1.1 OpenCV2开源视觉运算库	36
4.1.2 <i>SVM</i> ^{light} 运算库	36

目 录

4.1.3 STIP-2.0-linux	36
4.2 数据集合	37
4.3 Haar/AdaBoost行人检测	37
4.3.1 训练数据的准备	37
4.3.2 级联器训练	39
4.3.3 分类	40
4.4 HOG/SVM行人检测	44
4.4.1 OpenCV2相关库	44
4.4.2 CPU HOG简单例程	44
4.4.3 HOGDescriptor类接口	44
4.4.4 训练HOG人体特征模型	47
4.4.5 检测分类	48
4.5 STIP/SVM行为分析	48
4.6 行人检测结果	49
4.6.1 静态图像数据集测试	49
4.7 行为分析结果	50
第5章 总结与展望	51
5.1 总结	51
5.2 展望	51
参考文献	53
致 谢	56
外文资料原文	57
外文资料译文	61

主要符号表

符号	说明	页码
$\langle w, x \rangle$	内积运算	17
$ v _k$	k-范数	12

第1章 引言

1.1 背景介绍

机器视觉所研究的一个主要问题是：如何让机器视觉系统具备“计划”和“决策能力”？从而使之完成特定的技术动作（例如：智能汽车在检测到前方有行人时进行碰撞规避）。机器视觉系统作为一个感知器，为动作的决策提供信息。人体检测及行为分析是当前机器视觉领域的热点课题之一，对图像进行人体检测并对检测出的人类对象进行分析在诸多应用场景(智能汽车，高级机器人，监控系统)中是关键的一个环节。

从应用的角度来看，例如智能汽车这样的应用场景对人体检测及行为分析的性能要求非常严格(如智能汽车直接关乎驾驶环境下的安全程度)，实际应用中表现出来的性能指标稍微差一点便绝对不能投入实际使用。并且，机器视觉系统应该体现出相对于人类来完成某一工作时更加卓越的表现，如提高无人监控系统相对于传统手工登记的快速性，高级机器人相对于传统工种的敏锐性，智能汽车相对于手动驾驶的安全性等。所以，人体检测及行为分析是具有广阔前景同时非常具有挑战性的一个领域。

从机器视觉的角度来说，人体检测及行为分析是一项困难的任务。首先是因为显式模型的匮乏，机器很难以人的思维方式去对视野中的目标进行捕获和判断，机器学习技术选择从训练样本中学习隐式模型。其次是分类架构的性能问题，人体检测及行为分析从本质上来说是多级对象分类问题的一个案例，对特征模型进行低风险和高精确度的感知和理解进而做出分类需要鲁棒的分类架构。

1.2 研究现状

人体检测及行为分析在过去数年内吸引了相当数量的来自计算机视觉社区的研究兴趣。许多技术理论以特征模型和泛型架构的形式被提出，使得视觉识别领域从分类玩具对象实例发展到识别自然图像中的多种类的对象和场景，有了显著的进步，这受益于新的鲁棒的图像描述和分类方法。

另外，在数据集合方面，收集泛数据集合的工作量是相当大的，这在某种程度上造成了数据集合的匮乏(特别是实时和自然条件下的数据集合)。各种技术理论的实现离不开数据集合，数据集合的质量在很大程度上决定了该种技术架构能否达到理论提出的饱和性能。本节稍后部分将会对常用的数据集合进行概述。

1.2.1 主要部件

人体检测及行为分析系统的主要组件可以分为两部分：初始对象假设生成/兴趣区(ROI)选择，分类/模型匹配。

1.2.1.1 假设生成

获取初始对象假设生成最简单的方法是采用划窗技术，划窗技术设定检测窗口的尺寸和位置在图像上进行位移，这样往往会造成计算消耗超出处理容限[2]。通过基于已知的关于目标对象类的先验信息限制搜索空间或是将划窗技术与递增复杂度的级联分类器[4],[1]结合可以显著提高处理速度。

除了划窗技术外，其他获取初始对象假设的技术从图像数据中提取特征。例如在静态镜头的监督学习方法中通常会采用背景差分技术。另外一种技术采用兴趣点检测器[3]，这种技术受启发于通常产生于对象边界的图像亮度函数间断点所包含的大量信息。

1.2.1.2 模型匹配

获取初始对象假设之后，需要进行验证(分类)，这需要引入采用多种空域和时域线索的行人外貌模型，包括生成模型和判别模型[5]。生成模型和判别模型的主要区别在于后验概率估计方法。

生成模型 依据类条件密度函数 $p(X|Y)$ 对目标对象外貌进行建模，与类先验概率 $p(X)$ 联系起来采用贝叶斯方法可以推导出目标对象后验概率。

$$p(Y|X) = \frac{p(X|Y) \times p(Y)}{p(X)} \quad (1-1)$$

生成模型从统计的角度表述数据分布，反映同类数据本身的相似度，而不关心各个目标对象类之间的判别边界。

判别模型 与生成模型不同，判别模型直接近似估计贝叶斯最大后验概率进行决策，从训练实例中不同目标对象类之间获取判别函数的参数。这种方法不考虑样本的产生模型，即不关心训练数据本身的特性。直接研究预测模型，寻找不同

对象类之间的分类面，反映的是异类数据之间的差异。典型的判别模型包括k近邻算法，感知机，决策树，支持向量机等。

1.2.2 特征描述与提取算法

在像素强度上进行局部滤镜操作目前被广泛采用。非适应型Haar小波特征由Papagergiou 和Poggio用于描述图像。完备的特征词典代表了不同区域，尺寸以及方向的局部像素差异，其简洁性和快速性使得Haar小波特征得到普及。自动化的特征选择过程即多种AdaBoost类算法被用于选择最具区别的特征子集[1]，这实际上是一种针对分类任务的特征最优化。

类似的，其他特殊的空间特征架构被引入用于在训练过程中产生适应于底层数据的特征集合，即局部感知域[6]，受启发于人类视觉皮层的神经结构。近来的研究经验性地表明在人体分类方面自适应性的局部感知域相对于非适应性的Haar小波特征所具有的优越性。

采用其它思想的特征描述方法聚焦于之前提及的局部边沿结构模型图像亮度函数的间断性。从局部图像区域中计算图像梯度方向直方图并进行标准化在密集的[2](HOG)和稀疏的[7](SIFT)特征表达式中得到普及。密集的方向梯度直方图采用固定大小的图像单元进行计算。稀疏性特征描述方法采用兴趣点检测器进行预处理[7],[3]。

作为以上提到的空域特征的扩展，时空域特征得到提出。通过结合时域强度特征差异，Haar小波特征和HOG特征都可以得到扩展。报告显示时空域特征相对于空域特征具有优越性，但同时需要处理时域对齐问题。

1.2.3 学习算法

判别模型旨在从特征空间的模式类别中学习最优判决边界。从这个目标出发，在人体检测的背景下，多层神经网络通过调整网络参数来实现最小误差判据，被应用于自适应局部感知域特征[6]。另外，支持向量机[8],[9]已成为解决模式分类问题的有力工具，与神经网络相比，支持向量机最大化决策边界实现不同的目标对象类之间的区分度最大化，已被应用于和多类特征集合进行组合[2],[3]。非线性支持向量机相对于线性支持向量机，采用非线性核函数将样本映射到高维空间中，在获取性能提升的同时带来明显的计算消耗提升。

特征自动提取的AdaBoost算法[10]用于通过弱分类器的线性加权组合来构建强分类器，每个弱分类器对单一特征设置门限。Viola等人针对人体目标问题提出了改进的级联检测器[1]并得到许多人的改进，在训练过程中，每一层都聚焦于处理前一层的错误，随着级联器复杂度递增，分类精确度得到提升。

1.2.4 数据集合

数据集合规模是相当庞大的，通常包含数以千计的训练样本以及大量的测试图像。在人体检测及行为分析方面，数据集合在复杂度(动态变化的背景)和安全保护应用(碰撞或是行为风险评估)的情景真实性方面要求非常严格。

表 1-1 现有数据集合概览

数据集	训练样本	测试集合	说明
MIT CBCL	924(裁剪)	无	前后视角
INRIA Person	2416(裁剪)/1218(整图)	1132(裁剪)/453(整图)	彩色图像
Daimler Pedestrain	15660(裁剪)/6744(整图)	21790(整图)	车载采集标注
KTH Action	600(视频)		6类行为
Hollywood Human	233(自动)/219(手工)	211(手工)	采集32部电影

从表1-1中可看出，在人体检测方面，与其他数据集合相比，Daimler提供的行人数据库^①的大小和复杂度更能够使实验方面得出有意义的结论。训练图像在不同的时间和地点进行记录，除了人体都是保持直立姿势外，没有明亮度，姿势或是衣着方面的限制，图1-1给出了一些示例。



图 1-1 Daimler 行人数据库

在行为分析方面，多数已有的人类行为识别数据集合(如[11])只提供了处于控制和简化的场景设定下记录的较少的行为类别，这和现实生活应用要求的处理

^① 数据库免费提供给学术或是研究用途，可从<http://www.science.uva.nl/research/isla/downloads/pedestrians/index.html>下载。

包含具有个体差异的人类行为的自然视频有很大的差异，这些个体差异来自于表情，姿势，动作和衣着，透视效果和镜头运动，明亮度差异，以及场景遮挡变化等。[3]处理了当前数据集的限制问题，采集现实视频中的人类行为样本，引入自动标注电影中的人类行为的基于剧本对齐和文本分类的方法，图1-2给出了一些示例。



图 1–2 Hollywood 行为数据库三类样本：接吻，接电话，走出汽车

1.2.5 存在问题

评估基准 尽管技术理论的发展非常迅速，在实验方面，情况并不是那么乐观，不同技术架构的性能在不同的报告下差异巨大(甚至达到几个数量级)。这源自于缺少一个公认的评估基准。

维度灾难 特征矢量的维度上升的同时带来了明显的计算消耗和内存限制等问题，需要在足够描述外貌特征和实际可行性之间寻求权衡。

实际应用 综述[12]中报告的性能结果与实际应用要求相比，仍存在数量级层面的差异。这可能来自训练数据集的有限大小(是否足以覆盖特征空间)，测试场景的类型差异(背景，明亮度，分辨率)等。人体检测及行为分析系统方面还有相当多的工作需要完成。

1.3 本文研究内容

[12]报告指出，在人体检测方面，基于Haar小波特征的AdaBoost级联器[1]在低分辨率图像和(接近于)实时处理的条件下性能最优，而HOG特征和线性支持向量机的组合[2]在中等分辨率和低处理速度限制的条件下表现突出。本文将着重于研究Haar/AdaBoost和HOG/SVM架构在人体检测中的应用。

在行为分析方面，[3]处理了数据集合的限制，并将空域兴趣点扩展到时空域表示，在现有的数据库以及电影测试样本中表现突出，本文将着重于研究STIP/SVM架构在行为分析中的应用。

在具体实现方面，文章研究采用OpenCV2开源运算库[13]对研究的架构进行实现，参考了[14]提供的文档以及[15]书籍。

1.4 本文结构

文章分为以下几个章节：

第一章引言部分介绍课题研究背景，研究现状和存在的难题。其中研究现状部分中对具有代表性的特征模型和分类架构进行了概述。

第二章较为详细地介绍人体检测方面Haar/AdaBoost和HOG/SVM架构的基本原理，将分为特征描述(Haar 特征和HOG特征)以及分类架构(AdaBoost级联器和SVM)两个部分进行阐述。

第三章较为详细地对行为分析方面的时空兴趣点特征描述和SVM的组合的基本原理。同时，这一章将引入对样本自动提取技术详细描述。

第四章是一个关于具体实现的介绍。主要包含OpenCV2中提供的特征描述符的接口参数说明以及具体实现方法等，在章节末尾会简要呈现实现效果。

第五章总结了研究成果和实验经验，探讨了目前存在的问题和不足以及未来有可能需要完成的工作。

第2章 人体检测

2.1 代表性算法

根据[12]的报告指出，在人体检测方面，Haar/AdaBoost[1]在(接近于)实时处理速度限制和低分辨率图像条件下表现突出，而HOG/linSVM[2]在低处理速度限制和中等分辨率图像条件下性能获得明显提升。本章将着重研究以上两种架构在人体检测方面的基本原理和应用，结构上分为特征描述算法和分类算法两部分来分别进行阐述。

2.2 特征

2.2.1 Haar特征

早期图像描述采用了图像每一个像素点的强度，这种方法的明显缺点是计算消耗很大。Papageorgiou等人提出可以采用基于Haar小波的特征来在区域上进行操作[16],Paul Viola等人随后提出了Haar特征[17]，由于其快速性和简洁性，Haar特征迅速得到许多改进，Lienhart等人扩展了这个集合[18]，加入了倾斜的和中心环绕的特征子集，形成了一个过完备的特征字典(如图2-1所示)，包含了水平方向和垂直方向以及相应的倾斜的特征。

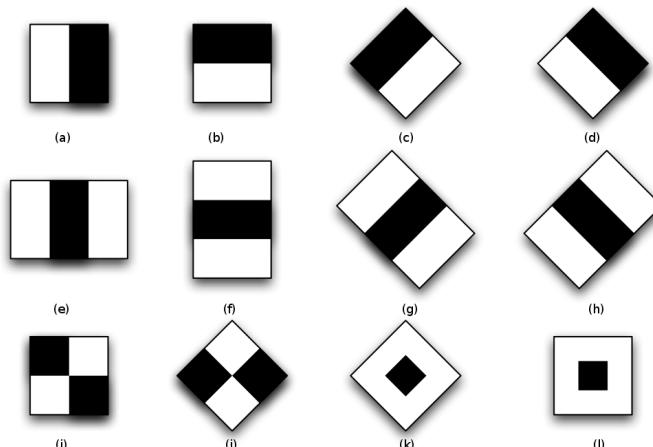


图 2-1 Haar小波特征。(a)-(d):边缘特征；(e)-(h):线特征；(i)-(j):角特征；(k)-(l):中心包围特征

图2-1中白色区域表示正区域 L , 黑色部分表示负区域 D 。某一区域上的特征值 v 通过如下方式计算:

$$v = \sum L - \sum D \quad (2-1)$$

图像中目标区域所具有的相似的统计特性经验性地表明这种特征计算方式能够比较好地描述目标特征(实际上是一个弱分类器)。但是简单地在图像上进行特征计算, 如此多的特征数量势必会带来计算消耗的问题。Viola等人采用积分图很好地加速了计算过程。积分图最早由Crow在1984年提出[19]。



图 2-2 Haar 特征的积分图加速计算

如图2-2所示, 设 $X(m, n)$ 为图像中的一个像素值, $L(X)$ 为 X 点所在的行从左端到 X 点扫过的像素值和, $I(X)$ 为 X 点和原点为对角点确定的矩形区域内的像素值和, 使用 $P(i, j)$ 表示位于坐标 (i, j) 处的像素值。

$$L(X) = \sum_{j=1}^{j=n} P(m, j) \quad (2-2)$$

$$I(X) = L(X) + L(m-1, n) \quad (2-3)$$

则图2-2中所示矩形 $ABCD$ 的像素和为:

$$\sum = I(C) + I(A) - I(B) - I(D) \quad (2-4)$$

这样只需对图像中的像素值进行一次扫描, 即可计算出所有需要的特征值。这是一种在线算法, 且对任何一个特征值的计算只需要常数时间。针对Lienhart提出的倾斜的Haar特征, 只需要引入倾斜的积分图进行扩展即可。

Haar小波特征提供了一种高效的划窗算法扩展, 每一个特征可以描述图像上特定特性的存在或者不存在, 比如边缘或者纹理的变化。这样的一个Haar特征是一个弱分类器, 其检测正确率比随机猜测强一些, 但不足以用于鲁棒地人体检测系统。稍后将介绍AdaBoost算法以及Viola-Jones目标检测框架, 引入复杂度递增

的退化决策树，形成一个强分类器。

2.2.2 HOG特征

HOG特征描述符由法国国家计算机技术和控制研究所的Dalal和Triggs提出[2]，其思想是一幅图像中的局部对象的外貌和形状可以被像素强度梯度或边缘的方向分布很好地描述。将图像划分为邻接的图像子区，称为胞元(cell)，然后对胞元内的每一个像素计算方向梯度直方图，最后将这些直方图联合起来形成最终特征描述符。

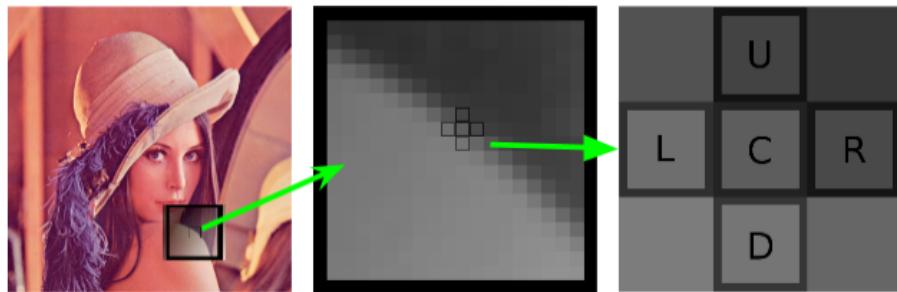


图 2-3 图像子区：胞元

梯度矢量(gradient vector) 梯度矢量是计算机视觉中的一个重要概念，许多视觉算法都需要引入对图像中每一个像素的梯度矢量的计算。如下图所示的 3×3 灰度图像(图2-3 中的一个胞元)，相应的像素标记字母作为标号。

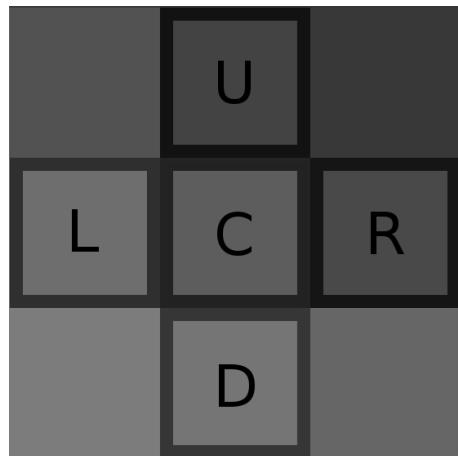


图 2-4 梯度矢量计算演示

像素值在0 – 255之间，0表示黑色，255表示白色。 $R - L$ 称为 x 方向变化率。需要注意的是，图中L像素的灰度值比R像素灰度值高，这样计算出来会是负值， $L - R$ 则是正值，也称为 x 方向变化率，但是一副图像中的计算方法应当保持一致。

类似的， $U - D$ 称为 y 方向变化率。两个方向的变化率取值在 $-255 \rightarrow 255$ 之间，编程实现上不能用一个字节存储，可以映射到 $0 \rightarrow 255$ 之间，这样，如果将变化率用灰度值表示，则非常大的负变化率将映射为黑色，非常大的正变化率将映射为白色。同时，我们可以得到一个梯度矢量 $[R - L, U - D]$ ，其幅度(magnitude)和相角(angle)计算方法如下：

$$\text{Magnitude} = \sqrt{(R - L)^2 + (U - D)^2} \quad (2-5)$$

$$\text{Angle} = \arctan\left(\frac{R - L}{U - D}\right) \quad (2-6)$$

如果采用带符号梯度($-255 \rightarrow 255$)，相角会分布在 $0^\circ \rightarrow 360^\circ$ 之间，如果采用无符号梯度(映射到 $0 \rightarrow 255$)，相角分布在 $0^\circ \rightarrow 180^\circ$ 之间。Dalal和Triggs发现使用无符号梯度在行人检测中表现更优[2]。

梯度矢量很好地提取了边缘信息。另一方面，试想将图像的明亮度提升，即：将图像中每个像素值加上同一个常数，重新计算梯度矢量会发现和明亮度变换之前的梯度矢量一致，这种性质使得梯度矢量可以被应用到特征提取中，即本文的人体特征提取中。

方向梯度直方图(Histogram of oriented gradient) 如下图的一个包含行人的图像，红色框标记一个 8×8 胞元，这些 8×8 的胞元将被用来计算HOG描述符。



图 2-5 密集胞元划窗

在每个胞元中，我们在每个像素上计算梯度矢量，将得到64个梯度矢量，梯度矢量相角在 $0^\circ \rightarrow 180^\circ$ 之间分布，我们对相角进行分箱(bin)，每箱 20° ，一共9箱([2]建议的最佳参数)。具有某一相角的梯度矢量的幅度按照权重分配给直方图。

这涉及到权重投票表决机制，Dalal和Triggs发现，采用梯度幅度进行分配表现最佳。例如，一个具有85度相角的梯度矢量将其幅度的1/4分配给中心为70°的箱，将剩余的3/4幅度分配给中心为90°的箱。这样就得到了下面的方向梯度直方图。

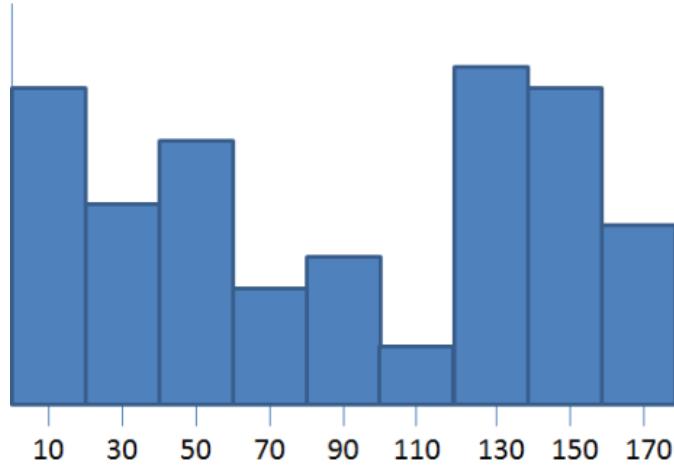


图 2-6 方向梯度直方图

上面分配幅度的方法可以减少恰好位于两箱边界的梯度矢量的影响，否则，如果一个强梯度矢量恰好在边界上，其相角的一个很小的扰动都将对直方图造成非常大的影响。同时，在计算出梯度后进行高斯平滑，也可以缓解这种影响。另一方面，特征的复杂程度对分类器的影响很大。通过直方图的构造，我们将特征[64个二元矢量]量化为特征[9个值]，很好地压缩了特征的同时保留了胞元的信息。设想对图像加上一些失真，对方向梯度直方图的扰动也不会太剧烈，这是HOG特征的优点。

前面提到，对图像所有像素进行加减后梯度矢量不变，接下来引入梯度矢量的标准化，使得其在像素值进行乘法运算后仍然保持不变。如果对胞元内的像素值都乘以某一常数，梯度矢量的幅度明显会发生变化，幅度会增加常数因子，相角保持不变，这会造成整个直方图的每个箱的幅度增加常数因子。为了解决这个问题，需要引入梯度矢量标准化，一种简单的标准化方法是将梯度矢量除以其幅度，梯度矢量的幅度将保持1，但是其相角不会发生变化。引入梯度矢量标准化以后，直方图各箱幅度在图像像素值整体乘以某个因子(变化对比度)时不会发生变化。

除了对每个胞元的直方图进行标准化外，另外一种方法是将固定数量的空域邻接的胞元封装成区块(block)，然后在区块上进行标准化。Dalal和Triggs使用 2×2 区块(50%重叠)，即 16×16 像素(如图2-7所示)。将一个区块内的四个胞元的直方图信息整合为36个值的特征(9×4)，然后对这个36元矢量进行标准化。

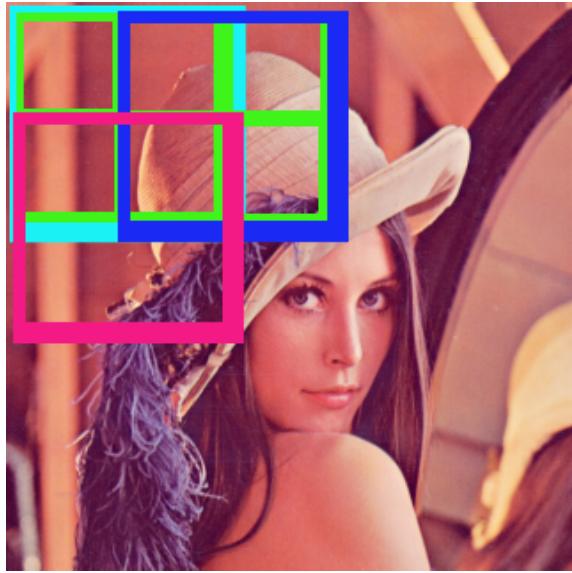


图 2-7 重叠区块

Dalal和Triggs在重叠区块设定下考察了四种不同的区块标准化算法，设 v 为未标准化的区块梯度矢量， $\|v\|_k(k=1,2)$ 是 v 的 k -范数(norm), e 是一个很小的常数(具体值并不重要)，其中三种标准化算法如下：

$$L2-norm : f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \quad (2-7)$$

$$L1-norm : f = \frac{v}{(\|v\|_1 + e)} \quad (2-8)$$

$$L1-sqrt : f = \sqrt{\frac{v}{(\|v\|_1 + e)}} \quad (2-9)$$

另外一种标准化算法*L2-Hys*是在*L2-norm*后进行截断，然后重新进行标准化。Dalal和Triggs发现*L2-Hys*,*L2-norm*,*L1-sqrt*性能相似，*L1-norm*性能稍有下降，但都相对于未标准化的梯度矢量有明显的性能提升。

区块重叠的影响是使得每个胞元会在最终得到的HOG描述符中其作用的次数大于1次(角胞元出现1次，边胞元出现2次，其它胞元出现4次)，但每次出现都在不同的区块进行重叠区块标准化。在划窗方法中定义一个区块位移的步长为8像素，则可以实现50%的重叠。

如果检测器窗口为64像素，则会被分为 7×15 区块，每个区块包括 2×2 个胞元，每个胞元包括 8×8 像素，每个区块进行9箱直方图统计(36值)，最后的总特征矢量将有 $7 \times 15 \times 4 \times 9 = 3780$ 个特征值元素。将HOG特征描述符递交给分类器进行训练，则可以实现特定的分类任务。

2.3 分类器

2.3.1 AdaBoost级联器

Valiant在1984年提出PAC(Probably Approximately Correct)可学习性，他认为“学习”是模式明显清晰或模式不存在时仍能获取知识的一种过程，并给出了一个从计算角度来获得这种过程的方法。PAC学习的实质是在样本训练的基础上，使学习算法的输出以概率接近未知的目标概念。PAC学习模型综合考察样本复杂度和计算复杂度将“学习”定义为形式化的概率理论。PAC学习模型涉及到两个重要的概念：弱学习和强学习。识别错误率小于 $\frac{1}{2}$ ，准确率仅比随机猜测略高的学习称为弱学习，识别准确率很高并能在多项式时间内完成的学习称为强学习。Valiant和Kearns提出了PAC学习模型中弱学习算法和强学习算法的等价性问题：任意给定仅比随机猜测略好的弱学习算法，是否可以将其提升为强学习算法？

基于PAC学习模型的理论分析，Schapire提出了Boosting算法[20]，对等价性问题做出了证明。Boosting算法的主要流程如下：

1. 从样本整体集合 D 中，不放回地随即抽样 $n_1 < n$ 个样本，得到集合 D_1 ，训练弱分类器 C_1
2. 从样本整体集合 D 中，抽取 $n_2 < n$ 个样本，其中合并进一半被 C_1 分类错误的样本，得到样本集合 D_2 ，训练弱分类器 C_2
3. 抽取 D 样本集合中， C_1 和 C_2 分类不一致样本，组成 D_3 ，训练弱分类器 C_3
4. 用三个弱分类器进行投票表决，得到最后分类结果

但是，这种算法存在实践上的缺陷，那就是都要求实现知道弱学习算法学习正确的下限即弱分类器的误差，另外Boosting算法可能会产生少数特别难区分的样本，导致不稳定问题。1995年，Freund和Schapire改进了Boosting算法，提出了AdaBoost(Adaptive Boosting)算法[10]，在效率几乎一样的情况下可以应用到实际问题中。

AdaBoost算法 给定样本集合： $(x_1, y_1), \dots, (x_m, y_m); x_i \in X, y_i \in Y, y_i = -1, +1$ ，初始化权重分布 $D_1(i) = 1/m$ ， $\sum d(x_i) = 1$ ，其算法流程如下，对 $t = 1, \dots, T$ ：

1. 使用权重分布 D_t 训练最弱分类器

2. 获取弱假设 $h_t : X \rightarrow -1, +1$, 错误率

$$\epsilon = Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$$

由于弱分类器比随机猜测略强, 我们期望 $\epsilon_t < 1/2$

3. 选取

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$

因为弱分类器的错误率 $\epsilon < 0.5$, 则有 $(1 - \epsilon_t)/\epsilon_t > 1 \Rightarrow \alpha_t > 0$,

4. 更新权重:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

Z_t 为标准化因子。 ϵ_t 越小, α_t 越大, 弱分类器 $h_t(x)$ 权重越大。

最后输出最终假设

$$H(x) = \operatorname{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$

从算法流程可以看出, AdaBoost 算法具有一些特点:

- 每次迭代改变的是样本的分布, 而不是重复采样
- 样本分布的改变取决于样本是否被正确分类, 分类错误的样本权值高
- 最终的结果是弱分类器的加权组合, 权值表示弱分类器性能

下面给出 AdaBoost 算法流程的可视化说明:^① 初始训练集合如下, 所有训练样本权重相等。

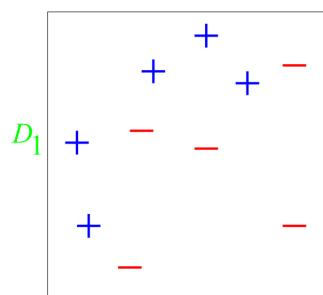


图 2-8 初始训练集合

^① 演示来自于Freund和Schapire写的“**A Tutorial on Boosting**”, <http://www.research.att.com/~yoav/>。

每次训练迭代根据错误率，得到新样本分布 D_{t+1} 以及该轮迭代获得的分类器 $h_t(x)$ 。圆圈标注表示误分类后权重增大，表示为新分布中较大的样本。

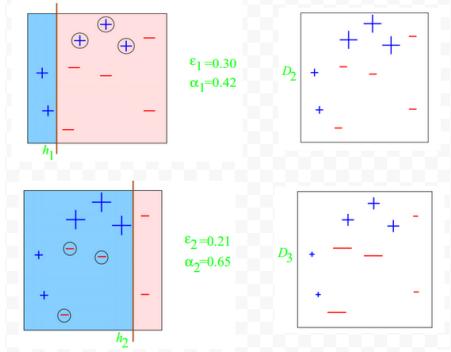


图 2-9 第1,2次训练

选择级联层数为3。

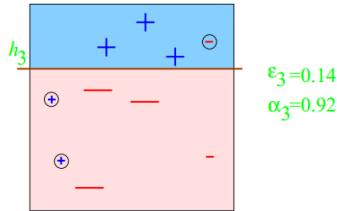


图 2-10 第3次训练

经过上一次迭代后可获得最终分类器：

$$\begin{aligned}
 H_{\text{final}} &= \text{sign} \left(0.42 + 0.65 + 0.92 \right) \\
 &= \text{sign} \left(1.99 \right) \\
 &= \boxed{\begin{array}{c|cc} + & + & - \\ + & - & - \\ + & - & - \end{array}}
 \end{aligned}$$

图 2-11 生成最终分类器

可以证明，AdaBoost算法随着迭代次数的增加，错误率上界会逐渐下降，另外即使训练次数很多，也不会出现过拟合的问题。Viola-Jones级联器框架利用AdaBoost的这些特性，训练复杂度递增的目标检测系统[1]，在每一个级联层，

AdaBoost算法被用于构建一个基于已选特征加权线性组合的分类器，使得包含人体和非人体样本的训练集合具有最低错误率。由于图像中大多数检测窗口都是非人体对象，级联器被调准来尽早地检测出所有行人同时排除非行人。由于前级的级联层快速排除非行人实例的过程中通常只有一小部分特征评估是必要的，这有助于级联器方法的快速性。

2.3.2 支持向量机(SVM)

支持向量机(SVM)是90年代中期发展起来的基于统计学习理论的一种机器学习方法，通过寻求结构化风险最小来提高学习机泛化能力，实现经验风险和置信范围的最小化，从而达到在统计样本量较少的情况下，亦能获得良好统计规律的目的。SVM最初由Vapnik提出，后来Cortes和Vapnik在1993提出了改进版本(软边界)并在1995年发表文章[21]进行了理论阐述。SVM从提出至今，在人体检测及行为分析系统中得到了许多应用[11],[2],[3]。

支持向量机涉及到比较多的统计和优化理论，理解起来比较困难，本节将从线性分类器(感知机)出发逐步解释其原理。^①

线性分类器(感知机) SVM实际上是一种感知机扩展。以二值分类器作为讨论对象，对于训练样本 $X_i = (x_{i1}, x_{i2}, \dots, x_{in}), Y_i = y_i, y_i \in \{-1, +1\}, i = 1, \dots, l$ ， X_i 为分类对象的特征向量，如前文介绍的HOG特征描述符矢量， Y_i 表征对象所属的类别。为了提高可视化程度可将特征向量用数据点来表示，如图2-12所示。

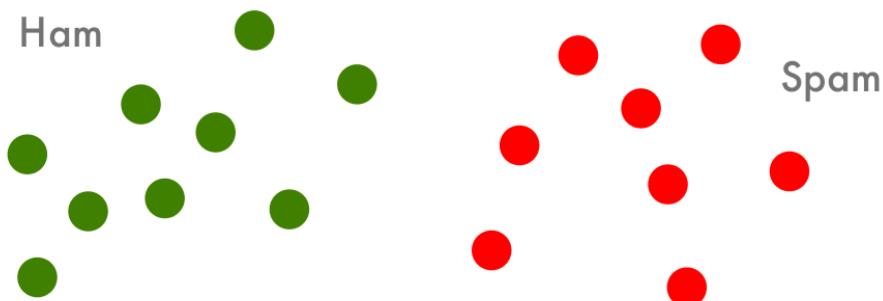


图 2-12 数据点的可视化表示

^① 本节的介绍思路和许多数学证明来自于斯坦福大学Andrew Ng讲授的CS229机器学习课程主页：<http://cs229.stanford.edu/>，以及卡耐基梅隆大学Alex Smola讲授的机器学习导论课程主页：<http://alex.smola.org/teaching/cmu2013-10-701/>

线性分类器需要在数据空间中寻找一个超平面(二维特征空间中表示为一条直线)，其方程可以表示为：

$$f(x) = \langle w, x \rangle + b \quad (2-10)$$

其中 w 称为权值矢量， b 为偏置量， $\langle w, x \rangle$ 为内积运算。如图2-13中的直线表示可能的线性分类器。

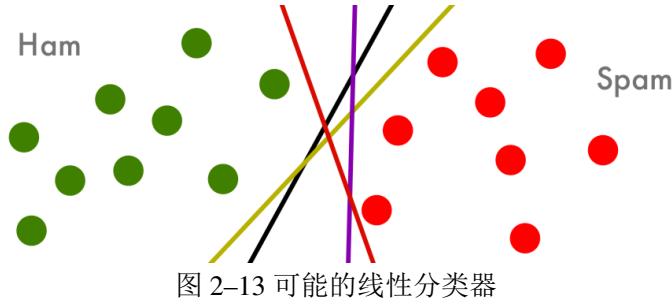


图 2-13 可能的线性分类器

Logistic回归 Logistic回归的目的是从特征学习出一个0-1分类模型，这个模型将样本特征的线性组合 $\theta^T x$ 作为自变量，由于自变量的取值范围是 $-\infty \rightarrow +\infty$ ，需要使用Logistic函数将自变量映射到区间 $(0, 1)$ 上，映射后的值被认为是判定为 $y = 1$ 的概率。可形式化表示为假设函数：

$$h_\theta(x) = g(\theta^T x) \quad (2-11)$$

x 是特征向量，函数 g 称为Logistic函数。为增加可视化程度，对于一元变量， $g(z) = \frac{1}{1+e^{-z}}$ 的函数图像：

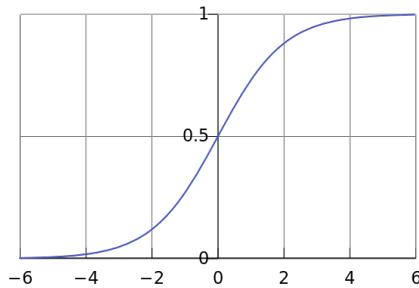


图 2-14 Logistic函数

从图2-14可以看到， $g(\cdot)$ 将 $(-\infty, +\infty)$ 映射到了 $(0, 1)$ ，给定 x ，其判定概率为

$$\Pr\{y = 1|x; \theta\} = h_\theta(x) \quad (2-12)$$

$$\Pr\{y = 0|x; \theta\} = 1 - h_\theta(x) \quad (2-13)$$

我们可对输入 x 预测一个输出 $y = 1$ 如果 $h_\theta(x) \geq 0.5$, 或等价地, $\theta^T x \geq 0$ 。考察一个阳性训练样本($y = 1$), $\theta^T x$ 越大, $h_\theta(x)$ 越大, 给样本判定 $y = 1$ 的可信度越高, 则若是 $\theta^T x \gg 0$ 我们能以非常高的可信度判定 $y = 1$ 。类似的, 当 $\theta^T x \ll 0$ 时我们可判定 $y = 0$ 。对于一个训练样本, 我们期望找到最佳的 θ 参数使得对于 $y_i = 1$ 的样本 $\theta^T x_i \gg 0$, 以及对于 $y_i = 0$ 的样本 $\theta^T x_i \ll 0$, 因为这样能够反映对训练集合中的样本数据非常好的拟合。

在SVM的讨论方面, 我们采用 $y \in \{-1, 1\}$ 来标注分类标签, 用 w, b 来替代 θ 参数, 将分类器表示为

$$h_{w,b}(x) = g(\langle w, x \rangle + b) \quad (2-14)$$

这里

$$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (2-15)$$

参数 w, b 表示可以显式地处理 b 参数, 令 $x_0 = 1$ (即对输入特征矢量增加一个维度)来将 $\theta^T x$ 表示为 $\langle w, x \rangle + b$, 即 b 取代 θ_0 , w 取代 $[\theta_1, \dots, \theta_n]^T$ 。

结构风险及泛化误差界 对一个分类器的性能评价分为两部分:

- **经验风险:** 表征分类器在给定训练样本上的误差
- **结构风险:** 表征对未知对象分类结果可信度, 体现泛化能力

这两者是相互制约的, 若是经验风险很小, 可能会产生过拟合状态, 未知样本与训练样本的微小差异都会导致分类错误, 即其泛化能力很差。相反, 设想结构风险很小, 则可能导致分类器将待分类的样本误判。泛化误差界可以综合表征经验风险和结构化风险, 可以表示对分类器经验风险和结构化风险之间的权衡。泛化误差界是由分类器的参数(即 w, b)唯一确定的, 接下来的问题是如何通过最优化 w, b 参数来最优化泛化误差界。在SVM中, 通常能做到经验风险接近于0, 则参数最优化的目的即是最小化结构化风险。我们再次给出线性分类器的一种演示

图2-15中 $\langle w, x \rangle + b = 0$ 表示判决边界(超平面), 比较理想的分类效果是

$$y = 1 \forall \langle w, x \rangle + b > 0 \quad (2-16)$$

$$y = 0 \forall \langle w, x \rangle + b < 0 \quad (2-17)$$

如何确定 w, b 呢? 我们经验性地指出通过寻找两条边界端直线之间实现最大间隔, 即最大间隔分类器。在给出证明之前, 我们先给出函数间隔与几何间隔的定义。

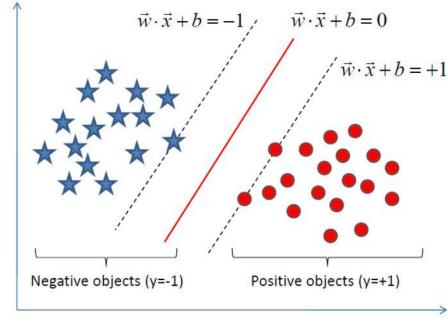


图 2-15 最佳参数的经验性演示

函数间隔与几何间隔 给定训练集合 $S = \{(x_i, y_i); i = 1, \dots, m\}$, 定义 (w, b) 关于某个训练样本的函数间隔为

$$\hat{\gamma}_i = y_i(\langle w, x \rangle + b) \quad (2-18)$$

定义 (w, b) 关于训练集合 S 的函数间隔为

$$\hat{\gamma} = \min_{i=1, \dots, m} \hat{\gamma}_i \quad (2-19)$$

一个较大的函数间隔 ($\hat{\gamma}_i(\langle w, x \rangle + b) > 0$) 表征可信度高或是正确的判决预测。则函数间隔在某种程度上可以衡量分类器的性能。但是, 设想将原有的 (w, b) 扩大为 $(2w, 2b)$, 这并不会改变分类器 $h_{w,b}(x)$ (仅取决于符号而与幅度无关), 但是函数间隔会扩大为原来的2倍, 这说明采用函数间隔不适合用于 (w, b) 的最优化衡量。这种情况下需要引入标准化方法。考察图2-16

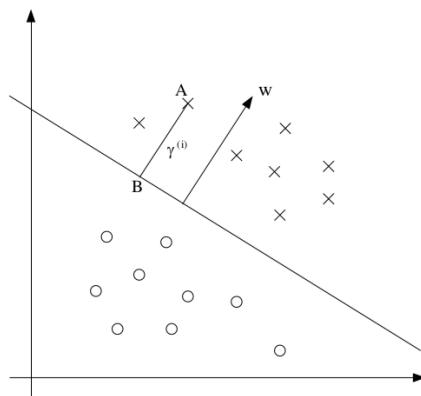


图 2-16 几何间隔演示

判决边界 $\langle w, x \rangle + b = 0$ 以及法向量 w 如图2-16中所示, w 与分类超平面垂直, 样本点 $A(x_i)$ 的位置在图中给出, 设其正确分类 $y = 1$, A 点距判决边界的距离由线段 AB 确定, 即 γ_i 。 B 点可由 $x_i(A$ 点)和 w 确定, 即: $x_i - \gamma_i \cdot w / \|w\|$ 。同时, B 点在

判决边界上，判决边界上的所有点 x 满足 $\langle w, x \rangle + b = 0$ ，则

$$\left\langle w, x_i - \gamma_i \frac{w}{\|w\|} \right\rangle + b = 0 \quad (2-20)$$

可解得

$$\gamma_i = \frac{\langle w, x_i \rangle + b}{\|w\|} = \left\langle \frac{w}{\|w\|}, x_i \right\rangle + \frac{b}{\|w\|} \quad (2-21)$$

更一般地，推广到适合阳性样本和阴性样本的表达式

$$\gamma_i = y_i \left(\left\langle \frac{w}{\|w\|}, x_i \right\rangle + \frac{b}{\|w\|} \right) = y_i \frac{\langle w, x_i \rangle + b}{\|w\|} \quad (2-22)$$

γ_i 称为关于样本的几何间隔。注意到如果 $\|w\| = 1$ ，则函数间隔和几何间隔相等。几何间隔独立于参数的尺度变化，即如果替换 (w, b) 为 $(2w, 2b)$ ，几何间隔并不会变化。与函数间隔的定义相似，对训练集合 $S = \{(x_i, y_i); i = 1, \dots, m\}$ ，定义 (w, b) 关于训练集合 S 的几何间隔为

$$\gamma = \min_{i=1, \dots, m} \gamma_i \quad (2-23)$$

最大间隔分类器 经过前面的讨论，我们提出最优化问题：

$$\max_{\gamma, w, b} \gamma \quad s.t. \quad y_i(\langle w, x_i \rangle + b) \geq \gamma, i = 1, \dots, m \quad \|w\| = 1. \quad (2-24)$$

即在满足每一个训练样本的函数间隔(γ_i)都至少是 γ 的条件下寻找最大的 γ 。 $\|w\| = 1$ 的限制保证函数间隔和几何间隔相等，这样每一个训练样本的几何间隔(对最优化有意义的)同样也至少是 γ ，即最优化问题的目的是求出具有最大几何间隔的 (w, b) 参数。因为有附加条件 $\|w\| = 1$ 的存在，原始最优化问题并不容易直接求解。注意到函数间隔与几何间隔之间的关系：

$$\gamma = \hat{\gamma} / \|w\| \quad (2-25)$$

我们可以将最优化问题转化为等价的：

$$\max_{\gamma, w, b} \frac{\hat{\gamma}}{\|w\|} \quad s.t. \quad y_i(\langle w, x_i \rangle + b) \geq \hat{\gamma}, \quad i = 1, \dots, m \quad (2-26)$$

为了进一步地求解，引入缩放限制：

$$\hat{\gamma} = 1 \quad (2-27)$$

即关于训练集合的函数间隔限定为1。因为 (w, b) 参数同时乘以一个尺度因子后函数间隔同时变化一个尺度因子，则实际上是对函数间隔进行缩放后 (w, b) 参数相应地变化一个尺度因子以寻求等效，而几何间隔对 (w, b) 的缩放是独立的，故这样的

限定并不影响最优化问题的求解。进一步地，可以发现最大化 $\hat{\gamma}/||w|| = 1/||w||$ 和最小化 $||w||^2$ 是等效的，可以得出如下的等效最优化问题：

$$\min_{\gamma, w, b} \frac{1}{2} ||w||^2 \quad s.t. \quad y_i(\langle w, x_i \rangle + b) \geq 1, \quad i = 1, \dots, m \quad (2-28)$$

此时，最优化问题实际上是一个凸优化问题，可以使用现有的二次规划(QP)程序进行求解。

然而，以上的讨论都是针对线性可分的数据进行的，对于线性不可分的数据，最优化问题根本不存在最优解，这也是线性分类器的瓶颈。为了解决线性分类器遇到的限制，引入了核函数。在了解核函数之前需要引入Lagrange对偶性问题，同时能获取一个比使用普通QP进行最优化更加高效的最优化求解方法。

Lagrange对偶性 考察下面的最优化问题：

$$\min_w f(w) \quad s.t. \quad h_i(w) = 0, \quad i = 1, \dots, l \quad (2-29)$$

在 $f(w)$ 为二次函数且约束条件为线性约束的时候，称为凸优化问题，可采用Lagrange算法求解。定义Lagrange函数为

$$\mathcal{L}(w, \beta) = f(w) + \sum_{i=1}^l \beta_i h_i(w) \quad (2-30)$$

β_i 称为Lagrange乘子。令

$$\frac{\partial \mathcal{L}}{\partial w_i} = 0; \quad \frac{\partial \mathcal{L}}{\partial \beta_i} = 0 \quad (2-31)$$

解出 w 和 β ，即是目标问题最优解。进一步地扩展约束条件，定义原始问题：

$$\min_w f(w) \quad s.t. \quad g_i(w) \leq 0, \quad i = 1, \dots, k; \quad h_i(w) = 0, \quad i = 1, \dots, l. \quad (2-32)$$

为了求解该问题，定义广义Lagrange函数：

$$\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^k \alpha_i g_i(w) + \sum_{i=1}^l \beta_i h_i(w) \quad (2-33)$$

α_i 和 β_i 称为Lagrange乘子。考察一个量

$$\theta_{\mathcal{P}}(w) = \max_{\alpha, \beta: \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta) \quad (2-34)$$

可以发现

$$\theta_{\mathcal{P}}(w) = \begin{cases} f(w) & \text{if } w \text{ satisfies primal constraints} \\ \infty & \text{otherwise} \end{cases} \quad (2-35)$$

当 w 满足约束条件的时候 $\theta_{\mathcal{P}}(w)$ 和目标函数相同，否则可发散至 ∞ 。这时，我们考

察问题

$$\min_w \theta_{\mathcal{P}}(w) = \min_w \max_{\alpha, \beta: \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta) \quad (2-36)$$

这和初始问题是等价的。令 $p^* = \min_w \theta_{\mathcal{P}}(w)$, 为初始问题的解。定义

$$\theta_{\mathcal{D}}(\alpha, \beta) = \min_w \mathcal{L}(w, \alpha, \beta) \quad (2-37)$$

我们可以提出对偶问题

$$\max_{\alpha, \beta: \alpha_i \geq 0} \theta_{\mathcal{D}}(\alpha, \beta) = \max_{\alpha, \beta: \alpha_i \geq 0} \min_w \mathcal{L}(w, \alpha, \beta) \quad (2-38)$$

将对偶问题的解标记为 $d^* = \max_{\alpha, \beta: \alpha_i \geq 0} \theta_{\mathcal{D}}(\alpha, \beta)$ 。可以发现如下关系

$$d^* = \max_{\alpha, \beta: \alpha_i \geq 0} \min_w \mathcal{L}(w, \alpha, \beta) \leq \min_w \max_{\alpha, \beta: \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta) = p^* \quad (2-39)$$

如果在某一条件下使得

$$d^* = p^* \quad (2-40)$$

则可通过求解对偶问题来获得初始问题的解。这样做的优点在于：一者对偶问题往往更容易求解；二者可以自然地引入核函数，进而推广到非线性分类问题。现在来考察初始问题和对偶问题解相同的条件。假设约束函数 f 和 g 都是凸的，并且存在 a_i, b_i ，使得 $h_i(w) = a_i^T w + b_i$ ，并且存在 w 使得 $g_i(w) < 0 \forall i$ ，这时，可以证明必然存在 w^*, α^*, β^* 使得 w^* 是初始问题的解，以及 α^*, β^* 是对偶问题的解，且 $p^* = d^* = \mathcal{L}(w^*, \alpha^*, \beta^*)$ 。同时， w^*, α^*, β^* 满足 Karush-Kuhn-Tucker(KKT) 条件：

$$\frac{\partial}{\partial w_i} \mathcal{L}(w^*, \alpha^*, \beta^*) = 0, \quad i = 1, \dots, n \quad (2-41)$$

$$\frac{\partial}{\partial \beta_i} \mathcal{L}(w^*, \alpha^*, \beta^*) = 0, \quad i = 1, \dots, l \quad (2-42)$$

$$\alpha^* g_i(w^*) = 0, \quad i = 1, \dots, k \quad (2-43)$$

$$g_i(w^*) \leq 0, \quad i = 1, \dots, k \quad (2-44)$$

$$\alpha^* \geq 0, \quad i = 1, \dots, k \quad (2-45)$$

反过来，如果 w^*, α^*, β^* 满足 KKT 条件，则其为初始问题和对偶问题的解。注意到方程(4-23)，表示如果 $\alpha_i > 0$ ，则 $g_i(w^*) = 0$ ，称为 KKT 对偶互补条件。这对 SVM 的支持向量的数量的证明以及 SMO 算法的收敛性非常重要。

分类器参数最优化 回顾之前提出的寻求最佳间隔分类器的最优化问题：

$$\min_{\gamma, w, b} \frac{1}{2} \|w\|^2 \quad s.t. \quad y_i(\langle w, x_i \rangle + b) \geq 1, \quad i = 1, \dots, m \quad (2-46)$$

可将约束条件重写为

$$g_i(w) = -y_i(\langle w, x \rangle + b) + 1 \leq 0, i = 1, \dots, m \quad (2-47)$$

由KKT对偶互补条件，仅当某一样本的函数间隔为1时 $\alpha_i > 0$ ($g_i(w) = 0$ 时)。考察图2-17，最大间隔分类器的判决边界用实线表示。

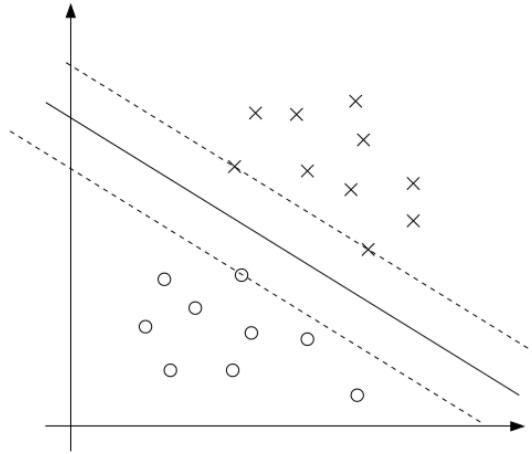


图 2-17 最佳间隔分类器演示

具有最小间隔的点相应地距离判决边界最近，如图2-17中位于虚线上的3个点。图中仅有3个点的 α_i 为非零值，称这些点为支持向量。事实上可以证明，支持向量的数量远小于训练集合的规模。

构建最优化问题的Lagrange函数如下

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^m \alpha_i [y_i(\langle w, x_i \rangle + b) - 1] \quad (2-48)$$

α_i 为Lagrange乘子。为了求得对偶问题，需要先求 $\mathcal{L}(w, b, \alpha)$ 关于 w 和 b 的最小值(固定 α)来求得 θ_D 。令

$$\nabla_w \mathcal{L}(w, b, \alpha) = w - \sum_{i=1}^m \alpha_i y_i x_i = 0 \quad (2-49)$$

可得到

$$w = \sum_{i=1}^m \alpha_i y_i x_i \quad (2-50)$$

同时令

$$\frac{\partial}{\partial b} \mathcal{L}(w, b, \alpha) = \sum_{i=1}^m \alpha_i y_i = 0 \quad (2-51)$$

代回Lagrange函数，得到

$$\mathcal{L}(w, b, \alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle - b \sum_{i=1}^m \alpha_i y_i \quad (2-52)$$

$$= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle \quad (2-53)$$

继而可求得对偶问题：

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle \quad (2-54)$$

$$s.t. \alpha_i \geq 0, i = 1, \dots, m; \sum_{i=1}^m \alpha_i y_i = 0 \quad (2-55)$$

满足 $p^* = d^*$ 需要的条件以及KKT条件，则我们可以通过求解对偶问题(SMO算法)来获得初始问题的解，即求得使 $W(\alpha)$ 在约束条件下最大化的 α ，然后采用方程(2-50)求得初始问题最优化的解 w ，然后可由

$$b^* = -\frac{\max_{i:y_i=-1} \langle w^*, x_i \rangle + \min_{i:y_i=1} \langle w^*, x_i \rangle}{2} \quad (2-56)$$

求得参数 b 。回顾之前的对偶问题，能够发现计算过程只涉及到求输入特征空间中的点之间的内积 $\langle x_i, x_j \rangle$ ，这是之后采用核函数的关键。

若已将SVM在训练集合上进行训练并得到了较好的拟合效果，需要对一个输入 x 进行分类预测，需要计算 $\langle w, x \rangle + b$ ，若是结果大于0，则输出 $y = 1$ 。由方程(2-50)，计算过程如下：

$$\langle w, x \rangle + b = \left\langle \sum_{i=1}^m \alpha_i y_i x_i, x \right\rangle + b \quad (2-57)$$

$$= \sum_{i=1}^m \alpha_i y_i \langle x_i, x \rangle + b \quad (2-58)$$

在 α_i 已知的情况下，为了进行一次判决，只需要计算 x 与训练集合中数据点之间的内积即可。前面已经提到除了支持向量外 $\alpha_i = 0$ ，这可以大幅减少计算量，只需要计算 x 与支持向量之间的内积即可。

核函数 上文针对SVM处理线性可分的情况，而对于非线性的情况，SVM的处理方法是选择一个核函数，通过将数据映射到高维空间，来解决在原始空间中线性不可分的问题。SVM的训练样本总是以成对内积的形式出现，且判决函数的表达式仅与支持向量的数量有关，而独立于空间的维度，在处理高维输入空间的

分类时，这种方法尤其有效。通过使用恰当的核函数来替代内积，可以隐式地将非线性的训练数据映射到高维空间，而不增加分类器参数规模。

在通过特征提取算法获得特征向量后，SVM可以采用 $\phi(x)$ 进行学习而不是原始特征值 x ，由于SVM算法的训练和分类的核心计算是求内积 $\langle x, z \rangle$ ，我们可以将其替换为 $\langle \phi(x), \phi(z) \rangle$ 。给定特征映射关系 ϕ ，定义核函数

$$K(x, z) = \langle \phi(x), \phi(z) \rangle \quad (2-59)$$

通常 $K(x, z)$ 的计算量会很小，甚至小于中间映射值 $\phi(x), \phi(z)$ 的计算量(高维空间)。选择一种高效的 $K(x, z)$ 计算方法，可以隐式地忽略 $\phi(x), \phi(z)$ 的计算。例如，给出核函数

$$K(x, z) = (\langle x, z \rangle + c)^d \quad (2-60)$$

特征将被映射到 $\binom{n+d}{d}$ 维特征空间，这很容易带来维数灾难。但是核函数以内积的形式在低维空间直接进行计算，而不用显式地写出映射后的结果，因而仍然能够在常数时间内完成。

$K(x, z)$ 可以理解为是 $\phi(x)$ 和 $\phi(z)$ 之间的相似度的衡量。当 x 和 z 相似时核函数接近于1，而当 x 和 z 正交时核函数接近于0。构造有效的核函数，成为接下来要解决的问题。给定有限训练集合 $\{x_1, \dots, x_m\}$ ，定义核函数矩阵为 $m \times m$ 矩阵 K ， $K_{ij} = K(x_i, x_j)$ ，这里不加证明地给出Mercer定理：

定理 2.3.1 如果函数 K 是 $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ 上的映射，那么如果 K 是一个有效核函数的充要条件是对于训练样本 $\{x_1, \dots, x_m\}$ ，其相应的核函数矩阵是半正定的。

同时，列举出一些常用的有效核函数模型：

- 线性核函数: $K(x_i, x_j) = \langle x_i, x_j \rangle$
- 多项式核函数: $K(x_i, x_j) = (\langle x_i, x_j \rangle + 1)^d$
- 径向基核函数: $K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{\sigma^2})$
- S型核函数: $K(x_i, x_j) = \tanh(\beta \langle x_i, x_j \rangle + r)$

松弛变量 如图2-18所示的线性可分样本，当加入一个扰动的样本后(称为离群点)，因为判定边界本身只有少数几个支持向量组成，离群点会造成分类器间隔明显缩小。虽然通过映射将低维线性不可分问题变为高维的线性可分问题，但是映射到高维空间后仍然可能存在离群点。这几乎难以避免，因为随着数据规模增大，样本数据本身由于随机性造成的偏差的扰动会变得明显。这种情况需要引入

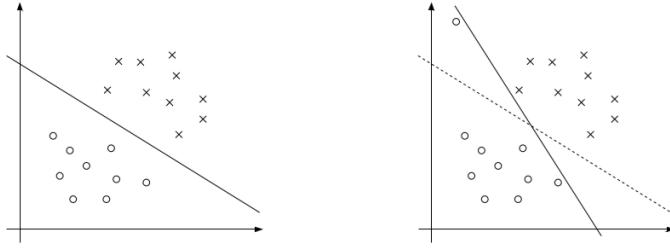


图 2-18 个别扰动对性能的影响

松弛变量 $\xi_i (i = 1, \dots, m)$, 对于分类任务, 不要求每个样本都分类正确, 允许存在误差。将最优化问题重写为

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \quad (2-61)$$

$$s.t. \quad y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \quad i = 1, \dots, m \quad (2-62)$$

$$\xi_i \geq 0, \quad i = 1, \dots, m \quad (2-63)$$

这样, 允许样本的函数间隔小于1, 若一个样本的函数间隔 $1 - \xi_i$, 我们需要通过给目标函数增加 $C\xi_i$ 来补偿离群扰动。惩罚因子 C 的作用是在寻找最优间隔超平面和数据点偏差量最小之间权衡, 表征对离群点的重视程度。

这时Lagrange函数为:

$$\mathcal{L}(w, b, \xi, \alpha, r) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i [y_i(x^T w + b) - 1 + \xi_i] - \sum_{i=1}^m r_i \xi_i \quad (2-64)$$

α_i 和 r_i 是Lagrange乘子。可得到对偶问题:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle \quad (2-65)$$

$$s.t. \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, m \quad (2-66)$$

$$\sum_{i=1}^m \alpha_i y_i = 0 \quad (2-67)$$

KKT对偶互补条件为:

$$\alpha_i = 0 \Rightarrow y_i(\langle w, x_i \rangle + b) \geq 1 \quad (2-68)$$

$$\alpha_i = C \Rightarrow y_i(\langle w, x_i \rangle + b) \leq 1 \quad (2-69)$$

$$0 < \alpha_i < C \Rightarrow y_i(\langle w, x_i \rangle + b) = 1 \quad (2-70)$$

剩下的问题就是如何求解对偶问题, 需要引入序列最小最优化(SMO)算法。

SMO算法 Platt在1998年提出SMO算法[22], 很快成为最快的二次规划优化算法, 特别针对线性SVM和稀疏数据时性能更优。对 α_i 进行优化的过程中, 注意

到 α_i 满足约束条件(2-67)，假设固定 $\alpha_2, \dots, \alpha_m$ ，对 α_1 进行优化，这实际上是不可行的，因为一旦更新 α_1 ，约束条件(2-67)则不满足了，因为

$$\alpha_1 = -y_1 \sum_{i=2}^m \alpha_i y_i \quad (2-71)$$

即 α_1 由 $\alpha_2, \dots, \alpha_m$ 唯一确定。如果要对 α_i 进行优化，至少要对两个参数同时进行优化以满足约束条件。出于这种动机，SMO算法按照如下流程迭代至收敛：

1. 采用启发式算法选择需要更新的一对 (α_i, α_j) ，使目标函数在该轮迭代能得到最大性能提升
2. 对 (α_i, α_j) 进行优化，保持其他所有的 $\alpha_k (k \neq i, j)$ 不变

为了测试SMO算法的收敛性，可以在收敛容限时间(TOL)下测试KKT条件是否满足，TOL通常设置在(0.001, 0.01)之间[22]。SMO算法的高效性在于每次更新 (α_i, α_j) 可以高效地进行。假设固定 $\alpha_3, \dots, \alpha_m$ ，在一次迭代中需要对 (α_1, α_2) 进行优化，有如下约束关系：

$$\alpha_1 y_1 + \alpha_2 y_2 = - \sum_{i=3}^m \alpha_i y_i = \zeta \quad (2-72)$$

可视化表示为：从图2-19中容易发现 α_1 和 α_2 必然被限定在 $[0, C] \times [0, C]$ 框中，同

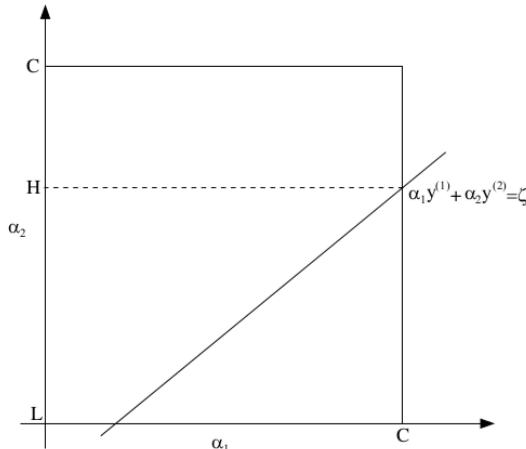


图 2-19 每一轮的迭代约束

时也被限定在直线 $\alpha_1 y_1 + \alpha_2 y_2 = \zeta$ 上，并且 $L \leq \alpha_2 \leq H$ ，否则 (α_1, α_2) 不能同时满足框限定和线限定。在示例中 $L = 0$ ，实际中 L, H 根据 $\alpha_1 y_1 + \alpha_2 y_2 = \zeta$ 表示的限定而不同。我们可以将 α_1 表示为 α_2 的函数：

$$\alpha_1 = (\zeta - \alpha_2 y_2) y_1 \quad (2-73)$$

重写目标函数为：

$$W(\alpha_1, \alpha_2, \dots, \alpha_m) = W((\zeta - \alpha_2 y_2)y_1, \alpha_2, \dots, \alpha_m) \quad (2-74)$$

在固定 $\alpha_3, \dots, \alpha_m$ 的情况下，这实际上是一个关于 α_2 的凸函数。在忽略边界框限定的情况下可以通过求偏微分进行凸优化，得到 α_2 优化后的值，用 $\alpha_2^{new,unclipped}$ 表示，然后根据 $[L, H]$ 进行截取：

$$\alpha_2^{new} = \begin{cases} H & \text{if } \alpha_2^{new,unclipped} > H \\ \alpha_2^{new,unclipped} & \text{if } L \leq \alpha_2^{new,unclipped} \leq H \\ L & \text{if } \alpha_2^{new,unclipped} < L \end{cases} \quad (2-75)$$

求得了 α_2^{new} 后，可代回求得 α_1^{new} 。

本章介绍了人体检测系统的代表性架构即其基本组件的基本理论，下一章将对行为分析方面的理论进行介绍。

第3章 行为分析

3.1 训练样本构建

数据集合对分类器的性能影响很大，前文的行人检测方面目前已有比较成熟的数据集合[12]，但是基于视频的行为分析方面数据集合存在很大限制。现有的数据集合[11]只提供处于人工控制和简化的场景设定下记录的数量较少的行为类别，这和实际应用中具有丰富现实性的待处理视频有很大差距。本文参考[3]提出的自动标注电影行为恢复方法，从电影中采集现实的自然的视频行为样本。

基于剧本的人类行为标注技术在本质上和近来一些采用文本信息从网络上进行自动图像采集[23]以及给图像中[24]和视频中[25]的角色自动命名的工具相似。不同的是这项工作将采用更加精细的文本分类工具来克服文本描述的类内差异。

电影里有多种类的和大量的现实人类行为，然而，往往一类行为在电影中只出现很少的次数，为了获取充足数量的行为样本用于视觉训练，对长时间电影片段进行标注是必要的，同时人工标注也是非常困难的一项任务。电影剧本在场景，角色，转录对白和人类行为等方面提供了详细的对电影内容的描述，包含有丰富的信息，并且许多电影的剧本是公开的。^①电影剧本中包含的丰富的信息已被Everingham等人用于视频中的角色自动命名[25]。这里我们扩展这种思想，应用基于文本的剧本搜索来自动地搜集人类行为视频样本。

然而，从剧本中自动标注人类行为同样面临许多困难的问题。首先，电影剧本通常没有时间信息，所以必须要和电影视频进行时域对齐。其次，剧本中描述的人类行为并不是总和电影中的行为相关。最后，自动行为恢复必须要处理文本中行为的大量类内的实质性的变化。

剧本和电影行为的对齐 电影剧本通常是普通文本格式并且具有相似的结构。我们利用行缩进作为简单特征来将剧本解析为独白，角色名称和场景描述等类别。为了将剧本和电影对齐我们采用[25]的方法并利用从互联网上现在的电影字幕中的时间信息。首先，我们采用单词匹配和动态变成对齐剧本和字幕中的对话片段。然后，将字幕中的时间信息转移到剧本中，并推测场景描述之间的时间间隔。如图3-1所示。

^① 可从www.dailyscript.com, www.movie-page.com, www.weeklyscript.com上获取。

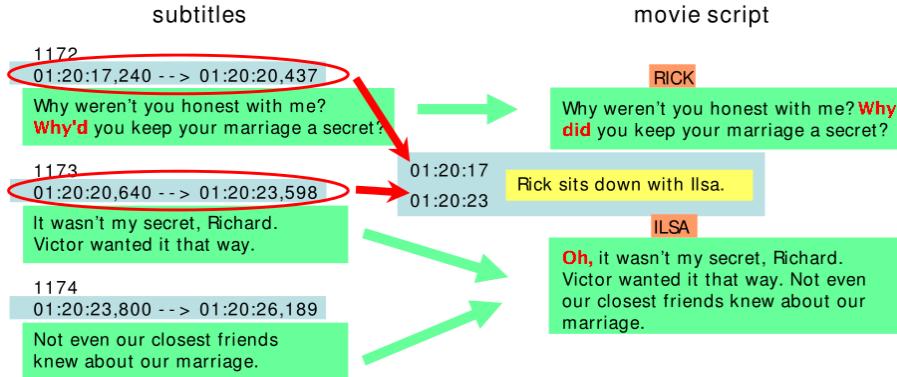


图 3-1 字幕和剧本中对话片段(绿色)的对齐演示。相邻对话片段的时间信息(蓝色标记)用来估计场景描述(黄色标记)的时间间隔

用于行为训练和分类的视频剪辑可以采用场景描述之间的时间间隔来定义，可能包含多个行为或是没有行为的片段。为了表征一个由剧本和字幕的不匹配造成的可能的不对齐的情况，我们将每一个场景描述和分数 a 关联起来。 a 通过匹配字数 $\#M$ 与邻接对话字数总量 $\#T$ 的比例计算：

$$a = \frac{\#M}{\#T} \quad (3-1)$$

时域不对齐可能由剧本和字幕的不完全匹配造成。而且，完美的字幕对齐 $a = 1$ 也不能保证正确的标注，因为剧本和电影之间可能还有差异。

基于文本的人类行为恢复 文本描述的人类行为表达方式会有一些类内差异，比如“走出汽车”这一行为，可能的表述有：

- Will gets out of the Chevrolet
- A black car pulls up.Two army officers get out
- Erin exits her new truck
- ...

并且，阳性误判不容易从阳性样本中分离出来，如“坐下”这一行为的两种不容易区分的误判：

- About to sit down,he freezes.
- He turned to sit down.But the smile dies on his face when he finds his place occupied by Ellie.

因此，基于文本的行为恢复并不是想象那么简单，通过简单的关键字搜索是很困难解决问题的。为了处理人类行为的类内变化，我们采用基于机器学习的文本分类方法[11]。分类器对剧本中的每个场景描述进行标记-包含目标行为或没有。

实现方法依赖于特征包模型[26]，每个场景描述用一个高维特征空间中的稀疏矢量表示。特征方面我们采用一个N个单词的窗口中的单词，邻接单词对以及非邻接单词对(N在2到8之间)。少于3个训练文档支持的特征被删除掉。分类方面采用和SVM等价的正则感知器[27]。分类器在人工标注的场景描述集合上进行训练，参数(正则常量，窗口尺寸N，判决门限)通过验证性集合进行调整。[3]报告指出文本分类器相对于简单的关键字匹配获得的明显性能提升。

Laptev等人[3]构建了两个视频训练集合^① (一个通过人工标注，一个自动标注)，以及一个视频测试集合。限定自动训练集合 $a > 0.5$ ，并将视频长度限定到1000帧以内。采用两个训练数据集，目的是对在监督学习设定和自动生成训练样本设定下分别进行评估。[3]指出自动化集合中正确标注视频的比例为60%，错误标注主要来自于剧本和视频不对齐以及少部分来自于文本分类器的错误，这些错误造成的影响会在之后讨论。

3.2 STIP特征

本文基于已存在的用于视频描述的特征包方法[28],[29],[30]并将静态图片分类中的经验扩展到视频中。Lazabnik等人[31]提出空域金字塔，即空域场景布局的粗描述法，能够提升识别性能。基于这种方法的一些成功的扩展包括单层金字塔的权重最优化[32] 以及广义空域网格[33]的采用。基于这些已有的方法，Laptev等人提出了构建空域-时域网格的方法[3]。

空域-时域特征 分析和解释视频行为特征近来在计算机视觉及其应用中吸引了许多关注，相对于静态图像，视频包含了关于场景变化的行为特征。传统的行为特征提取方法有光流(Barron等人1994年提出)和特征跟踪(Smith和Brady在1995年提出)。光流法通常只能捕获低阶行为，在行为迅速变化时容易失效；而特征跟踪法通常假设时域上恒定的外貌特征，在外貌发生变化时性能下降。

视频中的对象结构通常不是恒定的，而且视频正是因为对象结构的变化而表达出了非常丰富的信息。图像中局部空间和时间上像素值都有显著变化的点包含了关于行为的信息，这些点称为兴趣点。

稀疏表达的空域-时域特征近来表现出在行为识别方面的良好性能[30], [34],[29],[28]，这些方法提供了一种紧凑的视频描述并且对背景混杂，遮挡和尺寸变化具有耐受性。本文采用[35]，使用Harris操作的空域-时域扩展兴趣点。

^① 可从<http://www.irisa.fr/vista/actions> 下载。

空域兴趣点 在空域中，我们可以用线性尺度空间表达式来对图像进行建模 $f^{sp} : \mathbb{R}^2 \rightarrow \mathbb{R}$, 线性尺度空间 $L^{sp} : \mathbb{R}^2 \times \mathbb{R}_+ \rightarrow \mathbb{R}$

$$L^{sp}(x, y; \sigma_l^2) = g^{sp}(x, y; \sigma_l^2) * f^{sp}(x, y) \quad (3-2)$$

即 f^{sp} 和高斯核函数的卷积, $*$ 表示卷积运算, 高斯核函数:

$$g^{sp}(x, y; \sigma_l^2) = \frac{1}{2\pi\sigma_l^2} \exp(-(x^2 + y^2)/2\sigma_l^2) \quad (3-3)$$

Harris兴趣点检测的思想是寻找 f^{sp} 在每个方向都具有显著变化的点。给定 σ_l^2 , 这样的点可以采用一个整合了 σ_i^2 高斯窗口的二阶矩阵找到:

$$\mu^{sp}(\cdot; \sigma_l^2, \sigma_i^2) = g^{sp}(\cdot; \sigma_i^2) * ((\nabla L(\cdot; \sigma_l^2))(\nabla L(\cdot; \sigma_l^2))^T) \quad (3-4)$$

$$= g^{sp}(\cdot; \sigma_i^2) * \begin{pmatrix} (L_x^{sp})^2 & L_x^{sp} L_y^{sp} \\ L_x^{sp} L_y^{sp} & (L_y^{sp})^2 \end{pmatrix} \quad (3-5)$$

L_x^{sp} 和 L_y^{sp} 是在局部尺度 σ_l^2 上计算的高斯微分:

$$L_x^{sp} = \partial_x(g^{sp}(\cdot; \sigma_l^2) * f^{sp}(\cdot)) \quad (3-6)$$

$$L_y^{sp} = \partial_y(g^{sp}(\cdot; \sigma_l^2) * f^{sp}(\cdot)) \quad (3-7)$$

二阶描述符可理解为点的局部邻域图像定向的二维分布的协方差矩阵, 将其对角化处理, 得到两个特征值, 其不影响两个正交方向的变化分量。因此, μ^{sp} 的特征值 λ_1, λ_2 , ($\lambda_1 \leq \lambda_2$) 表征了 f^{sp} 在两个方向的变化。较大的 λ_1, λ_2 对应的即是兴趣点, 即在两个垂直方向像素强度都发生显著变化的点。

为了检测此类兴趣点, Harris 和 Stephens 在 1988 年提出了检测角点函数的极大值, 定义角点函数:

$$H^{sp} = \det(\mu^{sp} - k \operatorname{trace}^2(\mu^{sp})) \quad (3-8)$$

$$= \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad (3-9)$$

在兴趣点的位置上, 特征值的比率 $\alpha = \lambda_2/\lambda_1$ 较大(注意 $\lambda_1 \leq \lambda_2$)。对于 H^{sp} 的极大值, 比率 α 需要满足 $k \leq \alpha/(1 + \alpha)^2$, 如果设置 $k = 0.25$, H^{sp} 的正极值对应理想的各向同性的兴趣点, 即 $\alpha = 1, \lambda_1 = \lambda_2$, 稍低的 k 能够检测出具有更高 α 的更细长的兴趣点(在某一方向变化明显比另一个方向的变化显著)。常用的 k 值设定为 $k = 0.04$, 对应于检测 $\alpha < 23$ 的兴趣点。

时空域兴趣点 接下来, 我们将空域兴趣点扩展到时空域, 思想是检测局部时空卷中在时域和空域都发生显著变化的点, 具有此性质的点对应于特定位置的空

域兴趣点在时空卷中具有非匀速运动。我们采用函数 $f: \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ 并构建其线性尺度表达式 $L: \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}_+^2 \rightarrow \mathbb{R}$, 即将 f 和各向异性高斯核作卷积运算

$$L(\cdot; \sigma_l^2, \tau_l^2) = g(\cdot; \sigma_l^2, \tau_l^2) * f(\cdot) \quad (3-10)$$

高斯核具有空域方差 σ_l^2 和时域方差 τ_l^2 , 时空可分高斯核定义为

$$g(x, y, t; \sigma_l^2, \tau_l^2) = \frac{1}{\sqrt{(2\pi)^3 \sigma_l^4 \tau_l^2}} \times \exp(-(x^2 + y^2)/2\sigma_l^2 - t^2/2\tau_l^2) \quad (3-11)$$

时域上采用可分尺度参数是必要的, 因为时域和空域层面上的事件是独立的。与时域兴趣点处理相似, 我们采用一个时空域二阶矩矩阵, 即一个和高斯核相卷的 3×3 矩阵:

$$\mu = g(\cdot; \sigma_i^2, \tau_i^2) * \begin{pmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{pmatrix} \quad (3-12)$$

整合尺度 σ_i^2 和 τ_i^2 通过 $\sigma_i^2 = s\sigma_l^2$ 以及 $\tau_i^2 = s\tau_l^2$ 和 σ_l^2, τ_l^2 联系。类似地, 我们检测 f 中具有较大特征值 $\lambda_1, \lambda_2, \lambda_3$ 的区域, 为了实现检测, 我们扩展 Harris 角点方程为:

$$H = \det(\mu) = k \text{trace}^3(\mu) \quad (3-13)$$

$$= \lambda_1 \lambda_2 \lambda_3 - k(\lambda_1 + \lambda_2 + \lambda_3)^3 \quad (3-14)$$

定义 $\alpha = \lambda_2/\lambda_1$ 以及 $\beta = \lambda_3/\lambda_1$ 并重写 H :

$$H = \lambda_1(\alpha\beta - k(1 + \alpha + \beta)^3) \quad (3-15)$$

约束 $H \geq 0$, 可以得到 $k \leq \alpha\beta/(1 + \alpha + \beta)^3$, 当 $k = 1/27$ 时 $\alpha = \beta = 1$, 当 k 增大到充分大时, H 的局部正极大值对应于像素强度在时域和空域都具有显著变化的兴趣点。如果设定 α, β 的最大值为 23, 则 $k \approx 0.005$ 。图3-2是[35]给出的合成数据中得到的检测结果。

时空尺度因子的选择 从图3-2的(c),(d)可以看出, 时空域两个尺度因子的不同选择对实验结果有影响。[35]总结为: 时域内尺度因子越大, 表明动作发生的时间越短, 能够优先检测出动作持续时间短的特征点; 时域内尺度因子越小, 则优先检测动作持续时间长的特征点。Laptev等人通过去归一化后的在时间尺度和空间尺度 Laplace 算子最大值, 来检测时空域范围内事件, 基于这种机制能够得出尺度变换无关的时空兴趣点检测算子。

和[35]不同, 我们引入多尺度方法并在多层次空域-时域尺寸 (σ_i^2, τ_j^2) 上进行特征提取, $(\sigma_i^2, \tau_j^2); \sigma_i = 2^{(1+i)/2}, i = 1, \dots, 6; \tau_j = 2^{j/2}, j = 1, 2$ 。这样有利于减少计算复杂

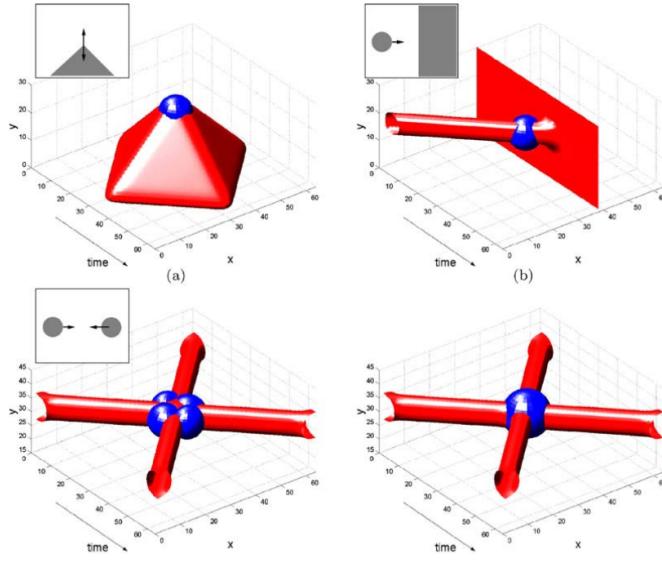


图 3-2 合成数据检测结果:(a)运动角;(b)运动球体和墙体的融合; (c)两个球体的碰撞, $\sigma_l^2 = 8, \tau_l^2 = 8$;(d)两个球体的碰撞; $\sigma_l^2 = 16, \tau_l^2 = 16$

度并消除人工选择尺寸的痕迹。

兴趣点区域特征化 为了使局部特征的动作和外貌特征化, 我们计算邻接检测点空域-时域卷的直方图描述符。每一卷的大小($\Delta_x, \Delta_y, \Delta_t$)和检测尺寸相关, $\Delta_x, \Delta_y = 2k\sigma, \Delta_t = 2k\tau$ 。每一卷被分为一个(n_x, n_y, n_t)的立方体网络, 对每一个立方体我们可以分别计算粗方向梯度直方图(HOG)和光流直方图(HOF), 然后进行标准化, 再通过多列索引串联为HOG或HOF特征描述符矢量。

时空特征包 由于人体的外观, 行为方式以及拍摄视角等存在差异, 对同一类动作在不同的视频中产生的兴趣点不尽相同, 但针对同一类动作, 这些兴趣点的特征具有相似性, 因此从兴趣点的特征集合中提取更高层, 能够代表相同动作的特征模式将有助于行为分析。

给定一个时空域特征集合, 我们构建一个时空域特征包(BoF), 这需要通过k-means算法进行聚类, 以构建视觉单词。BoF表达式将每一个特征分配给最近的(采用欧式距离)视觉单词, 并在时空域卷上计算视觉单词出现次数的直方图。时空域卷的定义与时空域网格的定义有关, 如果网格有多个子集, 不同子集的直方图串联成一个特征矢量后进行标准化。

空域维度方面[3]采用一个 1×1 网格(标准BoF表达式)以及一个 2×2 网格(表现突出), 一个水平 $h3 \times 1$ 网格以及一个垂直 $v1 \times 3$ 网格。同时, Laptev等人实现一个密集 3×3 网格以及一个邻接单元50%重叠的 $o2 \times 2$ 网格。时域维度方面, 将视频序列分为 t_1, t_2, t_3 三种分箱方式以及邻接单元重叠的 ot_2 分箱。和第二章中HOG描述符

的重叠一样，重叠部分的网格内位于中心的特征权重更大。这样，6种空域网格和4种时域网格可以组合成24中可能的时空域网格。图3-3描述了一些被证明是对行为识别有用的网格，每一个时空域网格使用HOG或是HOF描述符进行封装，称为通道，递交给分类器。

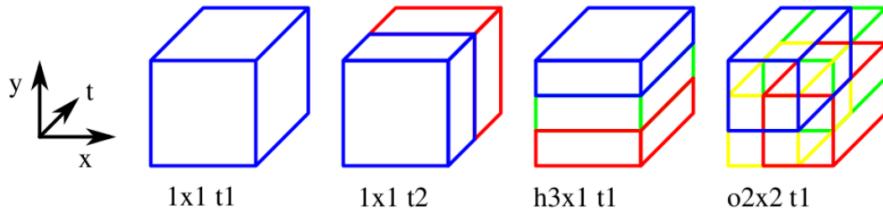


图 3-3 一些时空域网格示例

3.3 分类器

[3]采用非线性SVM进行分类任务，采用多通道高斯核

$$K(H_i, H_j) = \exp\left(-\sum_{c \in C} \frac{1}{A_c} D_C(H_i, H_j)\right) \quad (3-16)$$

$H_i = \{h_{in}\}$ 和 $H_j = \{h_{jn}\}$ 是 c 通道的直方图， χ^2 距离 $D_C(H_i, H_j)$

$$D_C(H_i, H_j) = \frac{1}{2} \sum_{n=1}^V \frac{(h_{in} - h_{jn})^2}{h_{in} + h_{jn}} \quad (3-17)$$

V 视觉单词的大小。 A_c 为 c 通道的所有训练样本之间的距离平均值。为了使SVM处理多类分类问题，最佳 C 集合通过贪心算法进行搜索，从空集合开始对增加或删除操作进行评估直到达到最优化。

和SVM进行二值分类不同，多类问题需要处理多个类别之间的间隔，第二章提到的最优化问题将急剧膨胀，计算量是不能忍受的。为了实现多类问题的求解，可以采用“一类对其余”(one-against-all)的方法，每次将一个类别定为正样本，将其余类别定为负样本，得到一个最优间隔分类器。比如一个5类问题，这样的方法最终会构建5个分类器，在需要对一未知数据进行分类的时候，需要遍历这5个分类器的结果。对于分类重叠问题，可以设置一个选择算法来，比如选择离超平面距离最远的一个判定作为结果。

第4章 具体实现及效果

4.1 运算库

4.1.1 OpenCV2开源视觉运算库

OpenCV[13]的全称是Open Source Computer Vision Library，是一个跨平台的计算机视觉库。OpenCV最早由Intel公司发起并参与开发，以BSD许可证授权发行，可以在商业和研究领域中免费使用。OpenCV可用于开发实时的图像处理，计算机视觉以及模式识别程序。最新的2.x版将开发和算法改用为C++接口。该程序库也可以使用Intel公司的IPP进行加速处理。

OpenCV具有长期保持活跃的用户社区以及由社区内的开发人员和用户维护的较完善的API文档[14]，同时也有关于OpenCV2 的书籍，如[15]等，总之，OpenCV2是目前最方便的开源视觉运算库。本文的部分实现都将依赖于OpenCV2提供的接口。

4.1.2 SVM^{light}运算库

SVM^{light}[36]是支持向量机的一种快速实现，主要由康奈尔大学计算机系的Thorsten Joachims负责的开发小组采用C编写，具有可调节内存需求的功能，可以用于许多二值分类问题。其主要特点是采用了多种快速优化算法提高程序的效率，支持标准核函数并可自定义配置。本文将采用SVM^{light}来实现HOG/linSVM架构。要实现多类问题的分类，需要使用其分支版本SVM^{struct}，即用于多类人体行为分析的分类任务中。

4.1.3 STIP-2.0-linux

由法国国家信息与自动化研究所的Laptev编写的用于计算视频中时空兴趣点的位置及特征描述符的程序，发布形式为二进制版本^①。程序基本算法来自[35]，尺度选择方面采用[3]中改进的多时空域尺度算法(参考第4章)。程序提供了检测兴趣点并可视化表示以及计算检测出的兴趣点的邻域(时空域网格)的HOG和HOF描述符的功能。

^① 可从www.di.ens.fr/~laptev/download.html下载。

4.2 数据集合

数据集合采用了[12]中用到的Daimler数据库，以及INRIA Person数据库等用于人体检测，行为分析方面采用[3]中采用自动标注方法提取的Hollywood Human Action数据库。关于数据集合的详细介绍见第1章。

4.3 Haar/AdaBoost行人检测

基于Haar小波的AdaBoost级联器[1]在低分辨率和(接近于)实时处理的应用场景下具有优势。本节介绍采用OpenCV2提供的运算库来对其进行实现的细节，主要涉及到训练数据的准备，训练以及分类的实现。

4.3.1 训练数据的准备

训练样本有两种类型：阴性样本(负样本)和阳性样本(正样本)。阴性样本没有包含目标对象，阳性样本则包含了待检测的对象。阴性样本必须手工准备，而阳性样本可以采用`opencv_createsamples`自动生成。

阴性样本 阴性样本可以从任意不包含待检测对象的图像中采样，阴性样本需要以特定格式列举在一个描述性的文本文件中，每一行包含一个文件名，需要注意的是样本中的图像分辨率需要大于训练窗口尺寸。描述文件的示例如下：

目录结构(阴性样本放置于`negative_images`文件夹内):

```
1 /negative_images  
2   img1.pgm  
3   img2.pgm  
4 negatives.txt
```

生成的文件列表描述文件`negatives.txt`格式:

```
1 negative_images/img1.pgm  
2 negative_images/img2.pgm
```

文档中要求手动生成，然而可以采用`bash`的`find`命令来自动生成文件列表描述:

```
1 find ./negative_images -iname "*.pgm" > negatives.txt
```

阳性样本 阳性样本通过`opencv_createsamples`来生成，可从单一图像或是经过预标记的图像文件中提取。阳性样本的数量依赖于特定应用，例如，在识别公司logo的应用中，可能只需要1个阳性样本，而在人脸识别或是人体识别中，需要数以千计甚至更多的样本。关于`opencv_createsamples`的参数说明：

- `-vec <vec_file_name>`:输出文件名
- `-img <image_file_name>`:源文件名
- `-bg <background_file_name>`:背景描述文件，用于对象随机失真背景
- `-num <number_of_samples>`:生成的阳性样本数量
- `-bgcolor <background_color>`:背景颜色(透明)，可以和`-bgthresh`配合设置背景色彩容限，在`bgcolor-bgthresh`和`bgcolor+bgthresh`区间内的像素视作透明.
- `-inv`:设置反色
- `-randinv`:随机反色
- `-maxidev <max_intensity_deviation>`:前景样本内像素的最大强度偏差
- `-max_x(y/z)angle <max_x(y/z)_rotation_angle>`: 最大旋转角度
- `-show`:调试选项，可以显示样本
- `-w <sample_width>`:输出样本的宽度
- `-h <sample_height>`:输出样本的高度

源图像会根据参数设置随机旋转，获得的图像随机放置在背景描述文件指定的任意背景上，按照参数设置的尺寸保存在`*.vec`文件中。阳性样本也可以从预标记的图像集合内获取，图像集合需要一个描述性的文本文件，每一行描述一个文件，以文件名开始，后面接对象数量和对象坐标((x, y, width, height)格式)。描述文件的示例如下：

目录结构:

```
1 /positive_images
2     img1.pgm
3     img2.pgm
4 positives.txt
```

生成的列表描述文件`positives.txt`文件格式:

```
1 /positives_images/img1.pgm 1 140 100 45 45
2 /positives_images/img2.pgm 2 100 200 50 50 50 30 25 25
```

从以上阳性样本集合中创建样本，需要`-info`参数:

- `-info <collection_file_name>`:描述文件名

不用设置失真，所以只还需要-w,-h,-show,-num等参数。

Daimler数据集内的正样本(DaimlerBenchmark/Data/TrainingData/Pedestrians)都是经过预标记裁剪的，所以只需要使用bash命令find可以生成阳性样本描述文件(以 18×16 分辨率为例):

```
1 find ./positive_images/ -name '*.pgm' -exec\  
2     echo \{\} 1 0 0 18 36 \; >positives.txt
```

然后可以进行样本创建:

```
1 opencv_createsamples -info positives.txt\  
2     -vec positives.vec -w 18 -h 36
```

创建完成后可以使用-show参数进行查看:

```
1 opencv_createsamples -vec positives.vec -w 18 -h 36
```

4.3.2 级联器训练

OpenCV提供了两种训练方法: `opencv_haartraining`和`opencv_traincascade`。后者是较新的版本，在OpenCV 2.x API框架下采用C++实现。`opencv_traincascade` 可以采用TBB库进行多线程运算,需要用TBB编译的OpenCV库。在训练完成后，级联器文件会保存在*.xml中。下面介绍关于`opencv_traincascade`的参数:

- `-data <cascade_dir_name>`:级联器保存参数
- `-vec <vec_file_name>`:前面得到的阳性样本文件名
- `-bg <background_file_name>`:背景文件(阴性)
- `-numPos(Neg) <numer_of_positive(negative)_samples>`:级联器每一层采用的阳性/阴性样本的数量
- `-numStages <number_of_stages>`:级联器级数
- `-precalcValBufSize <vals_buffer_size>`:预处理特征值的缓存区大小(Mb)
- `-precalclIdxBufSize <idxs_buffer_size>`:预处理特征值索引的缓存区大小(Mb),与训练速度正相关。
- `-baseFormatSave`:文件格式选择,指定后会存为旧格式
- `-stageType <BOOST(default)>`:层类型
- `-featureType <HAAR(default),LBP>`:特征类型, HAAR-Haar特征, LBP-局部二值特征

- `-w(h) <sampleWidth(Height)>`: 训练样本的尺寸, 必须与样本生成中采用的尺寸一致.
- `-bt <DAB,RAB,LB,GAB(default)>`: 级联类型: DAB-离散AdaBoost,RAB-Real AdaBoost,LB-LogitBoost,GAB-Gentle AdaBoost.
- `-minHitRate <min_hit_rate>`: 单级检测率要求
整体检测率大概为 $\text{min_hit_rate}^{\text{number_of_stages}}$.
- `-maxFalseAlarmRate <max_false_alarm_rate>`: 最大误判率要求, 整体误判率大概为 $\text{max_false_alarm_rate}^{\text{number_of_stages}}$.
- `-weightTrimRate <weight_trim_rate>`: 指定剪枝及权重, 建议选择为0.95.
- `-maxDepth <max_depth_of_weak_tree>`: 树的最大深度, 建议选择为1.
- `-maxWeakCount <max_weak_tree_count>`: 单级树数量
为了满足`-maxFalseAlarmRate`参数要求单级需要有 $\leq \text{maxWeakCount}$ 个树.
- `-mode <BASIC(default)—CORE—ALL>`: 选择Haar特征类型。 BASIC-采用垂直特征, ALL-采用所有特征(垂直和旋转, 如综述[12]中所示)。

[12]中指出级联层数 N_l 在 $N_l = 15$ 时达到饱和, 按照其参数选择, 在 18×36 阳性样本分辨率下, 配置15层级联, 采用所有Haar特征, 单层在15660个阳性样本和15660个阴性样本下训练, 选定单级50%的误判率和99.5%的检测率^①, 运行时特征值缓存区和特征值索引缓存区大小设置为1024MB和1024MB, 命令如下:

```

1 opencv_traincascade -data classifier -vec positives.vec \
2   -bg negatives.txt -numStages 15 -minHitRate 0.995 \
3   -maxFalseAlarmRate 0.5 -numPos 15660 -numNeg 15660 \
4   -w 18 -h 36 -mode ALL -precalcValBufSize 1024 \
5   -precalcIdxBufSize 1024

```

无论是社区还是实际操作来看, 训练数据的准备是比较快的(只涉及到转换为二进制文件, 计算消耗不大), 但是训练过程非常缓慢, 毕竟样本数量是非常庞大的, 采用AWS EC2来操作或许是一种快速可行的方法。

4.3.3 分类

在训练完成后, 利用获得的级联器*.xml文件, 可以进行分类(人体检测)测试。

^① 按照文档给定的估计方法, 整个15级系统的检测率为 $0.995^{15} = 0.9276$, 是比较低的, 社区内的代码建议为单级0.9999。

单个输入文件测试 程序实现读入单个测试文件并进行标记，在可视化输出预览的同时保存到输出文件。可以多次系统调用该程序实现对全部测试文件的标记。下面对程序的关键部分进行解释^①。

包含头文件，包括标准IO和OpenCV2的目标检测(objdetect)， GUI(highgui)以及图像处理(imgproc) 模块。声明命名空间。

```
1 #include "opencv2/objdetect/objdetect.hpp"
2 #include "opencv2/highgui/highgui.hpp"
3 #include "opencv2/imgproc/imgproc.hpp"
4
5 #include <iostream>
6 #include <stdio.h>
7
8 using namespace std;
9 using namespace cv;
```

全局变量，包括级联器文件名以及级联器和窗口ID。

```
1 String cascade_name = "cascade.xml";
2 CascadeClassifier ped_cascade;
3 string window_name = "Pedestrian_Detection";
```

将检测和显示功能封装进detectAndDisplay函数，提供给main函数调用。

```
1 void detectAndDisplay( Mat frame )
```

这样做的优点是模块化，并且方便未来工作的扩展。主要由detectMultiScale完成人体检测的工作，将检测到的人体区域存储进数组中。

```
1 ped_cascade.detectMultiScale( frame_gray, peds, 1.1, 2,
2                                     0|CV_HAAR_SCALE_IMAGE, Size(30, 30) );
```

然后对检测到的目标进行标记。

```
1 for( size_t i = 0; i < peds.size(); i++ )
2 {
3     Point upleft( peds[i].x, peds[i].y );
4     Point downright( peds[i].x + peds[i].width,
5                      peds[i].y + peds[i].height );
6     rectangle( frame, upleft, downright,
7                Scalar(255,0,0));
8 }
```

然后可视化输出并存储到文件。

^① 本论文中所有的实现都托管在<https://github.com/OnceMore2020/Thesis-Pedestrian.Detection.and.Activity.Recognition>，包括毕业设计期间的文献翻译，技术笔记等材料。

```
1 imshow( window_name , frame );
2 imwrite( "output.jpg" , frame );
```

主函数，加载级联器文件，读入输入图像，调用detectAndDisplay函数，完成对一张图片的检测。

```
1 int main( int argc , const char** argv )
2 {
3     Mat frame;
4
5     if( !ped_cascade.load( cascade_name ) )
6     { printf("----(!)Error loading\n"); return -1; }
7
8     frame = imread(argv[1]);
9     if( !frame.empty() ){
10         detectAndDisplay( frame );
11     }
12     else{
13         printf("----(!)Error reading image----Break!");
14         return -1;
15     }
16     waitKey(0);
17     return 0;
18 }
```

在Ubuntu12.04环境下使用CMake2.8编译并运行程序， CMakeList.txt内容如下：

```
1 cmake_minimum_required(VERSION 2.8)
2 project( HaarCascade )
3 find_package( OpenCV REQUIRED )
4 add_executable( HaarCascade haar.cpp )
5 target_link_libraries( HaarCascade ${OpenCV_LIBS} )
```

可选扩展：批处理输入文件 系统调用的方式是可以方便地控制处理文件的数量，以及随时可以选择暂停处理。缺点是不够自动化，设想程序能够自动读取文件夹内的输入文件，然后对所有文件进行批量识别，结果以文件的形式输出。现在来实现这个扩展。

首先设置存放测试文件的目录名字

```
1 static string testImagesFolder = "test/";
```

引入函数getFilesInDirectory，扫描目录内的所有文件，筛选出符合指定扩展的文件，然后存储进数组：

```
2 static void getFilesInDirectory(const string& dirName ,
3     vector<string>& fileNames , const vector<string>& validExtensions);
```

第4章 具体实现及效果

主函数main调用以上函数后逐个处理输入文件:

```
4     getFilesInDirectory(testImagesFolder, testImages, validExtensions);
5     unsigned long overallSamples = testImages.size();
6     cout << "Totally:" << overallSamples << "Files" << endl;
7
8     for(unsigned long tmp = 0; tmp < overallSamples; ++tmp){
9         const string currentImageFile = testImages.at(tmp);
10        frame = imread(currentImageFile);
11        if( !frame.empty() ){
12            detectAndDisplay( frame, currentImageFile );
13        }
14        else{
15            printf(" --(!) Error reading image -- Break!");
16            return -1;
17        }
18    }
```

注意detectAndDisplay函数参数列表扩展为两个，方便输入逐次覆盖源文件。

可选扩展：处理视频文件 实际应用中处理的多是实时视频文件，扩展程序能够读入视频文件，进行逐帧处理。首先指定输入文件名

```
19 String input_video="input.avi";      //input video file
```

然后读入视频文件，逐帧进行处理:

```
20     VideoCapture inputVideo(input_video);
21     if( inputVideo.isOpened() ){
22         double fps=inputVideo.get(CV_CAP_PROP_FPS);
23         cout << "FPS:" << fps << endl;
24         while(true)
25         {
26             bool bSuccess = inputVideo.read( frame );
27             if( !bSuccess )
28             {
29                 cout << "cannot get frames from video" << endl;
30                 break;
31             }
32             detectAndDisplay( frame );
33             if (waitKey(30)==27)
34             {
35                 cout << "Press ESC to exit" << endl;
36                 break;
37             }
38         }
39     }
40     else{
41         printf(" --(!) Error reading video -- Break!");
42         return -1;
43     }
```

4.4 HOG/SVM行人检测

4.4.1 OpenCV2相关库

OpenCV2在/samples/cpp/peopledetect.cpp中提供了采用HOG特征描述符实现的人体检测的例程，同时提供了GPU和OPENCL加速的HOG特征描述符，分别为gpu::HOGDescriptor和ocl::HOGDescriptor。CPU HOG的实现：<https://github.com/Itseez/opencv/blob/master/modules/objdetect/src/hog.cpp>。GPU加速的实现：</samples/gpu/hog.cpp>。OPENCL加速的实现：</samples/ocl/hog.cpp>。

OpenCV提供的头文件在include文件夹中，在Ubuntu 12.04下使用源码编译OpenCV2后可以在/usr/local/include文件夹下找到。

4.4.2 CPU HOG简单例程

HOGDescriptor类定义在object.hpp中，采用HOGDescriptor的实现可以在<https://github.com/Itseez/opencv/blob/master/samples/cpp/peopledetect.cpp>上找到。程序采用了HOGDescriptor::getDefaultPeopleDetector()来加载默认的行人检测器，接下来的问题是怎样使用手里已有的数据集来训练自己的分类器，这就需要了解OpenCV提供的HOG描述符类接口。

4.4.3 HOGDescriptor类接口

可以在objdetect.hpp中找到HOGDescriptor的类接口声明，在<https://github.com/Itseez/opencv/blob/master/modules/objdetect/src/hog.cpp>找到其实现。但是，OpenCV没有提供关于HOGDescriptor的文档，下面结合源码对重要的函数进行解释。

HOGDescriptor::HOGDescriptor 如下面代码所示：

```
1  CV_WRAP HOGDescriptor(Size _winSize, Size _blockSize, Size _blockStride,
2                         Size _cellSize, int _nbins, int _derivAperture=1, double _winSigma=-1,
3                         int _histogramNormType=HOGDescriptor::L2Hys,
4                         double _L2HysThreshold=0.2, bool _gammaCorrection=false,
5                         int _nlevels=HOGDescriptor::DEFAULT_NLEVELS)
```

结合第2章理论基础，很容易理解其参数列表：

- **_winSize** 检测器窗口尺寸，需要和区块尺寸和区块步进对齐。默认采用 64×128 像素。

- **_blockSize** 区块尺寸，需要和单元尺寸对齐。默认采用 16×16 像素。
- **_blockStride** 区块步进，需要是单元尺寸的整数倍。默认采用 8×8 像素。
- **_cellSize** 单元尺寸。默认采用 8×8 像素。
- **_nbins** 分箱数量。默认采用9箱。
- **_derivAperture** 微分算子。
- **_winSigma** 高斯平滑窗口参数。
- **_histogramNormType** 直方图标准化方式。默认采用L2-Hys标准化[2]。
- **_L2HysThreshold** L2-Hys标准化截断门限值。默认采用0.2。
- **_gammaCorrection** 标识是否要加入伽玛校正预处理模块。默认采用。
- **_nlevels** 多尺寸检测时HOG检测窗口最大缩放倍数。默认为64。

其中nlevels变量需要结合hog.cpp中的下面代码来理解其意思：

```

1 for( levels = 0; levels < nlevels; levels++ )
2 {
3     levelScale.push_back(scale);
4     if( cvRound(imgSize.width/scale) < winSize.width ||
5         cvRound(imgSize.height/scale) < winSize.height ||
6         scale0 <= 1 )
7         break;
8     scale *= scale0;
9 }
```

HOGDescriptor::getDescriptorSize 返回HOG特征描述符维度。

HOGDescriptor::setSVMClassifier 设置线性SVM分类器的参数。

HOGDescriptor::getDefaultPeopleDetector 返回OpenCV提供的用于人体检测的分类器参数(默认检测器窗口尺寸)。

HOGDescriptor::detect 进行对象检测，不进行检测窗口多尺寸缩放。

```

1 HOGDescriptor::detect(const Mat& img, CV_OUT std::vector<Point>& foundLocations,
2                         CV_OUT std::vector<double>& weights,
3                         double hitThreshold = 0, Size winStride = Size(),
4                         Size padding = Size(),
5                         const std::vector<Point>& searchLocations = std::vector<Point>())
```

参数列表：

- **img** 图像源文件，目前支持CV_8UC1和CV_8UC4格式的图像。
- **foundLocations** 检测到的目标对象边界左上角点的位置。
- **weights** 检测到的目标的可信度。
- **hitThreshold** SVM分类判定平面之间的距离门限。默认为0。

- **winStride** 窗口步进，需要是区块步进的整数倍。
- **padding** 图像边框尺寸
- **searchLocations** 划窗方法当前所在位置的边界左上角点的位置

程序先计算当前窗口的HOG特征描述符，然后计算距离，再与hitThreshold进行比较，如果大于hitThreshold 则存进foundLocations。

HOGDescriptor::detectMultiScale 进行多尺寸窗口目标对象检测。要检测多尺寸的目标，有两种方法：一是图像尺寸不变，缩放检测窗口大小，二是检测窗口不变，缩放图像尺寸。

```

1 HOGDescriptor::detectMultiScale(const Mat& img, CV_OUT vector<Rect>& foundLocations,
2                               CV_OUT vector<double>& foundWeights, double hitThreshold=0,
3                               Size winStride=Size(), Size padding=Size(), double scale=1.05,
4                               double finalThreshold=2.0, bool useMeanshiftGrouping = false)

```

与HOGDescriptor::detect相比增加的参数解释：

- **scale** 每次缩放的比例
- **finalThreshold** 聚类筛选门限
- **useMeanshiftGrouping** 聚类方法

结合下面的源码：

```

1 for( levels = 0; levels < nlevels; levels++ )
2 {
3     levelScale.push_back(scale);
4     if( cvRound(imgSize.width/scale) < winSize.width ||
5         cvRound(imgSize.height/scale) < winSize.height ||
6         scale0 <= 1 )
7         break;
8     scale *= scale0;
9 }

```

可知HOGDescriptor::detectMultiScale采用的是第二种方法。当图像缩小到比检测窗口小的时候就不再进行缩放了。当检测结束时，一些对象可能会被多个矩形窗口包围(检测到)，需要对这些窗口进行聚类分析。

```

1 if ( useMeanshiftGrouping )
2     groupRectangles_meanshift(foundLocations, foundWeights,
3                                 foundScales, finalThreshold, winSize);
4 else
5     groupRectangles(foundLocations, foundWeights, (int)finalThreshold, 0.2);

```

当useMeanshiftGrouping为true时，调用groupRectangle_meanshift进行聚类，否则调用groupRectangle进行聚类(默认采用)。groupRectangle函数对所有输入矩形采用矩形相似标准(相似尺寸和位置)进行聚类，最后一个参数0.2表示聚类时的相似度判决参数。然后某些些矩形类内矩形数量小于等于finalThreshold的矩形类被排除。然后将输出存进foundLocations。

HOGDescriptor::compute 返回对整个图像计算得到的HOG描述符。用于分类器的训练。

```
44 HOGDescriptor::compute(const Mat& img,
45                         CV_OUT vector<float>& descriptors,
46                         Size winStride=Size(), Size padding=Size(),
47                         const std::vector<Point>& locations=std::vector<Point>())
```

参数列表：

- **img** 源文件
- **descriptors** 存储HOG描述符

4.4.4 训练HOG人体特征模型

下载SVM^{light}程序包后解压至工作目录，下面来介绍分类器的实现。

训练HOG描述符 在工作目录下设置文件夹/pos和/neg分别用于放置阳性样本和阴性样本。

```
1 static string posSamplesDir = "pos/";
2 static string negSamplesDir = "neg/";
```

程序扫描目录内的文件，需要用到Haar/AdaBoost实现部分的批量处理扩展。对于扫描到的每一个图像文件，计算其HOG特征描述符矢量，将计算过程封装进calculateFeaturesFromInput函数中，其函数原型为：

```
1 static void calculateFeaturesFromInput(const string& imageFilename,
2                                         vector<float>& featureVector,
3                                         HOGDescriptor& hog);
```

接下来调用SVM^{light}，进行训练，将得到的SVM模型存储到文件。

```
1 SVMlight::getInstance()->read_problem(const_cast<char*> (featuresFile.c_str()));
2 SVMlight::getInstance()->train();
3 SVMlight::getInstance()->saveModelToFile(svmModelFile);
```

根据训练得到的支持向量(仅其对检测有用，第2章)，生成单个HOG特征模型。

4.4.5 检测分类

进行分类时，只需要加载训练得到的特征模型，就可以进行目标检测。将检测过程封装进detectTest函数中

```

48 static void detectTest(const HogDescriptor& hog, Mat& imageData) {
49     vector<Rect> found;
50     int groupThreshold = 2;
51     Size padding(Size(32, 32));
52     Size winStride(Size(8, 8));
53     double hitThreshold = 0.; // tolerance
54     hog.detectMultiScale(imageData, found, hitThreshold, winStride,
55                         padding, 1.05, groupThreshold);
56     showDetections(found, imageData);
57 }
```

4.5 STIP/SVM行为分析

STIP特征提取方面采用Laptev等人编写的软件包进行提取，并计算检测到的兴趣点时空域邻域的时空特征描述符。和[35]采用的自适应缩放不同，使用[3]中提出的采用一系列固定大小的多尺寸检测，这样的简化在实际中能够减少计算复杂度并保持几乎相似的性能。

特征描述符方面采用了HOG和HOF描述符，计算每一个兴趣点邻域内的特征矢量，时空域网格采用 $3 \times 3 \times 2$ 的分块，HOG特征采用4-分箱，HOF特征采用5-分箱，将每个分块的特征矢量计算出来之后进行串联，分别形成72元和90元描述符。下面对STIP的参数进行解释。

输入/输出 对于输入视频文件，可以指定帧处理间隔，输出特征点的位置和计算出的特征矢量。也可以人为指定兴趣点位置，通过文本文件传递兴趣点参数，格式如下：

```
1 # point-type x y t sigma2 tau2 detector-confidence
```

point-type可以是任意整数，sigma2和tau2为时空域窗口尺寸参数，根据[3]中的方法，可取值为 $\sigma^2 = \{4, 8, 16, 32, 64, 128, 256, 512\}$; $\tau^2 = \{2, 4\}$ 。

输出文件内的特征存储格式为：

```
1 # point-type y_norm x_norm t_norm y x t sigma2 tau2 descriptor
```

参数解释

- -i 指定输入文件，可选指定开始帧和结束帧
- -vpath 视频文件路径
- -ext 视频文件后缀，默认为avi
- -fpath 人工预指定的兴趣点描述文件
- -o 存储输出特征的文件名
- -det 特征检测方法，默认为harris3d[35]
- -dscr 描述符类型，可选hog,hof,hog三种类型
- -vis [yes/no] 开启或关闭可视化，默认开启
- -stdout [yes/no] 输出为标准库支持还是OpenCV库支持，默认关闭

根据[35],[3]中的参数建议，我们选择时空域 $3 \times 3 \times 2$ 网格，特征描述符采用HOG计算，兴趣点检测采用Harris角点检测函数扩展，这样能够得到STIP特征描述符，输出到samples-stip.txt文件。

```
1 ./bin/stipdet -i ./data/video-list.txt -vpath ./data/ -o ./data/samples-stip.txt
2         -det harris3d -dscr hog
```

将特征描述符递交给SVM分类器进行训练，得到的模型即可用于行为分析任务。

4.6 行人检测结果

4.6.1 静态图像数据集测试

Haar/AdaBoost架构的部分输出图像如下(INRIA Person数据集)：从结果可以

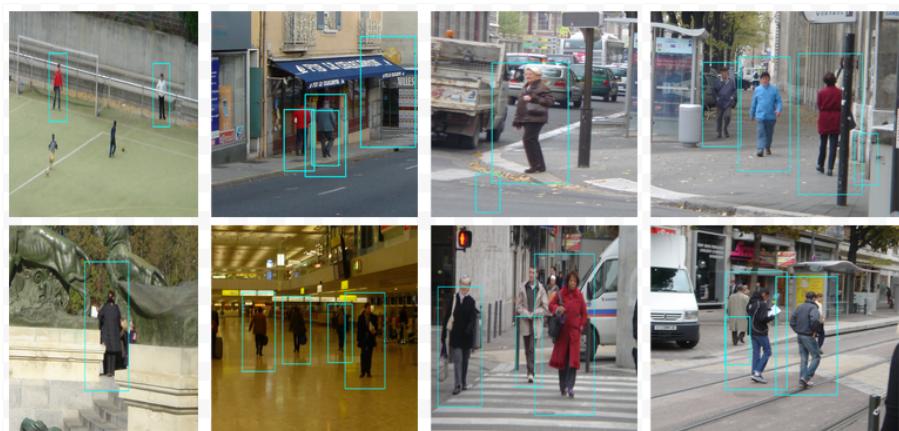


图 4-1 haar/AdaBoost输出结果

看出，识别出了图片中的大部分人体，但是仍然存在少数不能识别的目标以及

误判。另外一个问题时边框重叠的问题，可通过对结果进行筛选聚类来消除，这在HOG/linSVM中得到了体现。

HOG/linSVM架构识别结果(采用相同的输入，用绿色框标记):

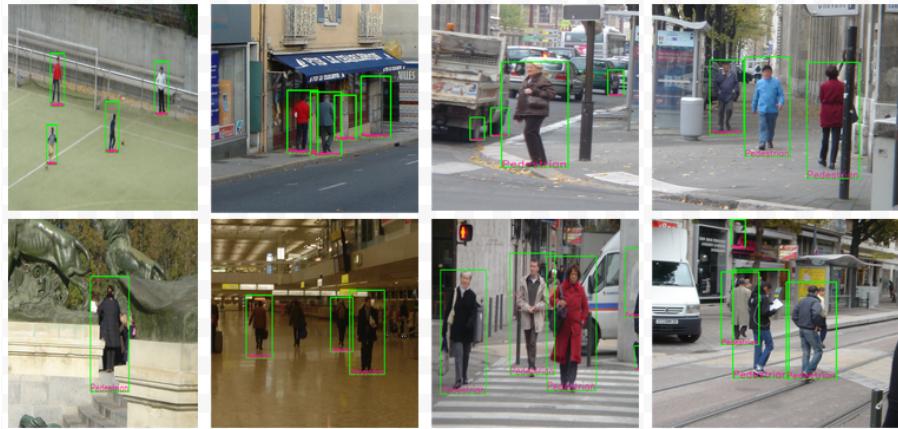


图 4-2 HOG/linSVM输出结果

在相同的输入下，HOG/linSVM的识别精度更高，但是误判率也有所提高。

4.7 行为分析结果

行为分析针对视频输入文件，经实验得到，对于较高分辨率的视频，处理速度无法达到实时处理的效果，将分辨率调小能够提高一定的实时性。检测效果抽样输出：



图 4-3 行为分析输出采样

第5章 总结与展望

5.1 总结

首先，本文归纳了具有代表性的系统(尤其是人体检测系统)具有代表性的算法架构的理论，并辅以必要的数学模型的推导。然后，基于OpenCV2以及一些运算库对选择的方法进行了简单实现。在人体检测方面，Haar/AdaBoost[1]架构能够在实时处理条件下实现较好的识别，HOG/linSVM在识别精度方面表现突出，但是实时性不够突出。两种架构都能达到90%以上的命中率。在行为分析方面，借助Laptev等人公布的时空兴趣点[3]程序包计算出的HOG描述符特征，递交给SVM进行训练，能够实现基本的行为捕捉识别。

对于实验结果的分析，能够得到以下几点重要结论：

- **参数调整** 对同一种架构，参数的选择对最后识别效果的影响非常大，这些参数主要包括划窗缩放因子，划窗尺寸，目标尺寸预判(最小目标，最大目标等)，而这些参数都比较依赖于采用的图像数据分辨率。
- **数据集合** 良好的训练数据对基于机器学习的视觉任务至关重要。
- **SVM核函数** 在运用SVM学习分类时，常选择的核函数是径向基核函数，能取得相对于线性核函数更好的性能，但是同时会带来计算消耗的上升。
- **误判目标** 从识别效果中的误判可以看出，人体作为任务目标时，竖直的结构更容易发生误判(遮阳伞，栏杆等)。
- **丢失目标** 未检测出来的目标往往表现出和背景融合，以及人体之间的相互遮挡。另外一些未检测到的目标表现出和训练样本中不同的站立姿势，这也是依赖于训练数据集的。

5.2 展望

在算法架构方面，HOG特征得到了许多完善并趋于成熟，例如[4]在HOG特征提取算法的前提下，采用一个Fisher弱学习器进行学习，然后采用AdaBoost算法构成分类架构，取得了更好的识别精度以及实时性。STIP特征在得到提出时[3]采用了稀疏的表达式来进行描述，之后的改进提出了密集的描述方式。

在具体实现方面，我们在上一节总结了存在的主要问题等，现在来探讨一些解决方法作为未来工作的起点。

- **参数动态选择** 对于不同的算法架构，我们尝试取得其最佳工作状态的参数，然后对输入的图像进行参数方面的归一化，这样做的目的是使算法能够相对独立于输入图像分辨率的影响。
- **预处理模块** 误判和丢失目标的原因都是复杂变化的背景造成的，引入一些预处理模块作为系统前级，或许能够在很大程度上改善识别效果，比如背景差分，直方图匹配等。
- **实时处理** 要提高实时处理性能，需要从减少计算复杂度入手。总结采用的方法可以发现，划窗方法以及特征计算耗时突出。可以通过引入一些先验知识(相机视角，肤色模型)等来进行目标预筛选，让分类器处理更有可能是目标的对象区域，这样能够提高实时性能。

参考文献

- [1] P. Viola, M. J. Jones, D. Snow. Detecting pedestrians using patterns of motion and appearance[J]. International Journal of Computer Vision, 2005, 63(2):153–161.
- [2] N. Dalal, B. Triggs. Histograms of oriented gradients for human detection[C]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005:886–893.
- [3] I. Laptev, M. Marszalek, C. Schmid, et al. Learning realistic human actions from movies[C]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2008:1–8.
- [4] I. Laptev. Improvements of object detection using boosted histograms[C]. British Machine Vision Conference. 2006:949–958.
- [5] I. Ulusoy, C. Bishop. Generative versus discriminative methods for object recognition[C]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005, 2:258–265.
- [6] C. Wohler, J. Anlauf. An adaptable time-delay neural-network algorithm for image sequence analysis[J]. IEEE Transactions on Neural Networks, 1999, 10(6):1531–1536.
- [7] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91–110.
- [8] V. N. Vapnik. The Nature of Statistical Learning Theory[M]. Springer-Verlag, 1995.
- [9] C. M. Bishop. Pattern Recognition and Machine Learning[M]. Springer, 2007.
- [10] Y. Freund, R. E. Schapire. A Decision-theoretic Generalization of On-line Learning and an Application to Boosting[J]. Journal of Computer and System Sciences, 1997, 55(1):119–139.
- [11] C. Schuldt, I. Laptev, B. Caputo. Recognizing human actions: a local SVM approach[C]. International Conference on Pattern Recognition. 2004, 3:32–36.
- [12] M. Enzweiler, D. Gavrila. Monocular Pedestrian Detection: Survey and Experiments[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(12):2179–2195.
- [13] OpenCVDevTeam. Open Computer Vision Library[DB/OL]. 2014. <http://opencv.org/>.
- [14] OpenCVDevTeam. OpenCV Documentation[DB/OL]. 2014. <http://docs.opencv.org/>.
- [15] R. Laganière. OpenCV 2 Computer Vision Application Programming Cookbook[M]. Packt Publishing Limited, 2011.
- [16] C. Papageorgiou, M. Oren, T. Poggio. A general framework for object detection[C]. Sixth International Conference on Computer Vision. 1998:555–562.

- [17] P. Viola, M. Jones. Rapid object detection using a boosted cascade of simple features[C]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2001, 1:511–518.
- [18] R. Lienhart, J. Maydt. An extended set of Haar-like features for rapid object detection[C]. International Conference on Image. 2002, 1:900–903.
- [19] F. C. Crow. Summed-area Tables for Texture Mapping[J]. Special Interest Group on GRAPHics and Interactive Techniques, 1984, 18(3):207–212.
- [20] R. E. Schapire. The Strength of Weak Learnability[J]. Machine Learning, 1990, 5(2):197–227.
- [21] C. Cortes, V. Vapnik. Support-Vector Networks[J]. Machine Learning, 1995, 20(3):273–297.
- [22] Sequential minimal optimization: A fast algorithm for training support vector machines[J]. Microsoft Research Technical Report, 1998.
- [23] L. jia Li, G. Wang, L. Fei-fei. Optimol: automatic online picture collection via incremental model learning[C]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2007.
- [24] T. Berg, A. Berg, J. Edwards, et al. Names and faces in the news[C]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2004, 2:848–854.
- [25] M. Everingham, J. Sivic, A. Zisserman. “Hello! My name is... Buffy” – Automatic Naming of Characters in TV Video[C]. British Machine Vision Conference. 2006.
- [26] F. Sebastiani. Machine Learning in Automated Text Categorization[J]. ACM Computing Surveys, 2002, 34(1):1–47.
- [27] K.-Y. K. Wong, T.-K. Kim, R. Cipolla. Learning Motion Categories using both Semantic and Structural Information[C]. IEEE Conference on Computer Vision and Pattern Recognition. 2007:1–6.
- [28] F. Schroff, A. Criminisi, A. Zisserman. Harvesting Image Databases from the Web[C]. IEEE International Conference on Computer Vision. 2007:1–8.
- [29] M. Marszałek, C. Schmid, H. Harzallah, et al. Learning Object Representations for Visual Object Class Recognition[J]. Visual Recognition Challange workshop, 2007.
- [30] P. Dollar, V. Rabaud, G. Cottrell, et al. Behavior recognition via sparse spatio-temporal features[C]. Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance. 2005:65–72.
- [31] S. Lazebnik, C. Schmid, J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories[C]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2006, 2:2169–2178.

参考文献

- [32] A. Bosch, A. Zisserman, X. Munoz. Representing Shape with a Spatial Pyramid Kernel[C]. ACM International Conference on Image and Video Retrieval. 2007.
- [33] R. Lienhart. Reliable Transition Detection in Videos: A Survey and Practitioner's Guide[J]. International Journal of Image and Graphics, 2001, 1:469–486.
- [34] C. W. J. W. M. Everingham, L. van Gool, A. Zisserman. Overview and results of classification challenge[C]. The PASCAL VOC'07 Challenge Workshop. 2007.
- [35] I. Laptev. On Space-Time Interest Points[J]. International Journal of Computer Vision, 2005, 64(2-3):107–123.
- [36] Joachims T. SVMLight[DB/OL]. 2014. <http://svmlight.joachims.org/>.
- [37] L. G. Valiant. A Theory of the Learnable[J]. Communications of the ACM, 1984, 27(11):1134–1142.
- [38] N. Cristianini, J. Shawe-Taylor. An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods[M]. Cambridge University Press, 2000.

致 谢

首先感谢张翔老师在毕业设计期间的指导，他悉心推荐了优秀的参考文献，并帮助我开拓思路，给出了非常宝贵的建议。他是一个真正的学者。感谢计算机视觉社区里杰出科学家和工程师的刻苦工作，他们始终对机器视觉的进展保持着敏锐洞察力，他们提出的模型和架构，是人类智慧和文明的结晶。感谢Knuth教授设计的TeX和Bram Moolenaar开发的VIM。这些工具都是上帝对工科男的馈赠。最后，向我的父母致敬，谢谢他们的养育，教会我学以济世的道理。

Monocular Pedestrian Detection: Survey and Experiments

Markus Enzweiler, Dariu M. Gavrila^①

Abstract

Pedestrian detection is a rapidly evolving area in computer vision with key applications in intelligent vehicles, surveillance, and advanced robotics. The objective of this paper is to provide an overview of the current state of the art from both methodological and experimental perspectives. The first part of the paper consists of a survey. We cover the main components of a pedestrian detection system and the underlying models. The second (and larger) part of the paper contains a corresponding experimental study. We consider a diverse set of state-of-the-art systems: wavelet-based AdaBoost cascade, HOG/linSVM, NN/LRF, and combined shape-texture detection. Experiments are performed on an extensive data set captured onboard a vehicle driving through urban environment. The data set includes many thousands of training samples as well as a 27-minute test sequence involving more than 20,000 images with annotated pedestrian locations. We consider a generic evaluation setting and one specific to pedestrian detection onboard a vehicle. Results indicate a clear advantage of HOG/linSVM at higher image resolutions and lower processing speeds, and a superiority of the wavelet-based AdaBoost cascade approach at lower image resolutions and (near) real-time processing speeds. The data set (8.5 GB) is made public for benchmarking purposes.

1.1 Introduction

Finding people in images is a key ability for a variety of important applications. In this paper, we are concerned with those applications where the human body to be detected covers a smaller portion of the image, i.e., is visible at lower resolution. This covers outdoor settings such as surveillance, where a camera is watching down onto a street,

① 文献引用标号以原文为准。

or intelligent vehicles, where an onboard camera watches the road ahead of possible collisions with pedestrians. It also applies to indoor settings such as a robot detecting a human walking down the hall. Hence our use of the term “pedestrian” in the remainder of the paper, rather than the more general “people” or “person.” We do not consider more detailed perception tasks such as human pose recovery or activity recognition.

Pedestrian detection is a difficult task from a machine vision perspective. The lack of explicit models leads to the use of machine learning techniques, where an implicit representation is learned from examples. As such, it is an instantiation of the multiclass object categorization problem (e.g., [79]). Yet the pedestrian detection task has some of its own characteristics, which can influence the methods of choice. Foremost, there is the wide range of possible pedestrian appearance, due to changing articulated pose, clothing, lighting, and background. The detection component is typically part of a system situated in a physical environment, which means that prior scene knowledge (camera calibration, ground plane constraint) is often available to improve performance. Comparatively large efforts have been spent to collect extensive databases; this study, for example, benefits from the availability of many thousands of samples. On the other hand, the bar regarding performance and processing speed lies much higher, as we will see later.

Pedestrian detection has attracted an extensive amount of interest from the computer vision community over the past few years. Many techniques have been proposed in terms of features, models, and general architectures. The picture is increasingly blurred on the experimental side. Reported performances differ by up to several orders of magnitude (e.g., within the same study [74] or [39] versus [74]). This stems from the different types of image data used (degree of background change), the limited size of the test data sets, and the different (often, not fully specified) evaluation criteria such as localization tolerance, coverage area, etc.

This paper aims to increase visibility by providing a common point of reference from both methodological and experimental perspectives. To that effect, the first part of the paper consists of a survey, covering the main components of a pedestrian detection system: hypothesis generation (ROI selection), classification (model matching), and tracking.

The second part of the paper contains a corresponding experimental study. We evaluate a diverse set of state-of-the-art systems with identical test criteria and data sets as follows:

- Haar wavelet-based AdaBoost cascade;
- histogram of oriented gradient(HOG) features combined with a linear SVM;
- neural network using local receptive fields; and
- combined hierarchical shape matching and texture-based NN/LRF classification

In terms of evaluation, we consider both a generic and an application-specific test scenario. The generic test scenario is meant to evaluate the inherent potential of a pedestrian detection method. It incorporates no prior scene knowledge as it uses a simple 2D bounding box overlap criterion for matching. Furthermore, it places no constraints on allowable processing times (apart from practical feasibility). The application-specific test scenario focuses on the case of pedestrian detection from a moving vehicle, where knowledge about camera calibration, location of the ground plane, and sensible sensor coverage areas provide regions of interest. Evaluation takes place in 3D in a coordinate system relative to the vehicle. Furthermore, we place upper bounds on allowable processing times (250 ms versus 2.5 s per frame). In both scenarios, we list detection performance both at the frame and trajectory levels.

The data set is truly large-scale; it includes many tens of thousands of training samples as well as a test sequence consisting of 21,790 monocular images at 640×480 resolution, captured from a vehicle in a 27-minute drive through urban traffic. See Table 1. Compared to previous pedestrian data sets, the availability of sequential images means that also hypothesis generation and tracking components of pedestrian systems can be evaluated, unlike with [28], [46], [49]. Furthermore, the data set excels in complexity (dynamically changing background) and realism for the pedestrian protection application onboard vehicles.

The scope of this paper is significantly broader than our previous experimental study [49], which focused on pedestrian classification using low-resolution pedestrian and non-pedestrian cutouts (18×36 pixels). Here, we evaluate how robust and efficient pedestrians can be localized in image sequences in both generic and application-specific (vehicle) settings. Among the approaches considered, we include those that rely on coarse-to-fine image search strategies, e.g., see Section 4.4.

The remainder of this paper is organized as follows: Section 2 surveys the field of monocular pedestrian detection. After introducing our benchmark data set in Section 3, Section 4 describes the approaches selected for experimental evaluation. The result of

the generic evaluation and the application-specific pedestrian detection from a moving vehicle are listed in Section 5. After discussing our results in Section 6, we conclude the paper in Section 7.

单眼视觉行人检测:综述和实验

Markus Enzweiler,Dariu M. Gavrila^①

摘要

行人检测是计算机视觉中快速发展的一个领域,在智能汽车,监控系统和高级机器人等方面具有关键性应用.这篇文章的目的是同时从方法学和实验学视角提供一个关于(该领域)目前的技术发展水平的综述.文章的第一部分是一个概览.这一部分涵盖了行人检测系统的主要组件和底层模型.文章的第二(同时也是占更大比重的)部分是一个相关的实验研究.我们考察了目前具有代表性的多种系统模型:基于小波的AdaBoost级联器[74], HOG/linSVM[11],NN/LRF[75],和联合形状-纹理检测器 [23].实验采用城市环境行驶车辆捕获的泛数据集.数据集包含了多达数以千计的训练样本以及一个27分钟的包含了超过20,000张具有行人位置注释的图像的测试序列.我们考察了一般评估设定和车载系统行人检测的特殊评估设定.实验结果表明HOG/linSVM在高分辨率和低处理速度条件下的明显优势,同时,基于小波的AdaBoost级联器在较低分辨率和(接近于)实时处理速度条件下的优势.数据集(8.5GB)公诸于众满足基准测试的目的.

1.1 引言

对图像进行人体检测是诸多重要应用的关键环节.在这篇文章中,我们只关注那些待检测人体只占图像较小部分的应用设定,即在低分辨率下的可视对象.这包括了诸多户外设定,例如:摄像头俯视监视街道的监控系统,车载摄像头监视前方道路的行人以评估潜在碰撞可能性的智能汽车.人体检测同时也可应用于诸如机器人检测过道上行人的室内设定.因此本文剩余部分我们都用“行人”这个词,而不是更泛义的“人”.我们不考察例如人类姿态复原或是行为识别等更具体的任务.

行人检测从机器视觉的角度来说是一项困难的任务.由于显式模型的匮乏,我们选择使用从实例样本中学习隐式表示的机器学习技术.就其本身而言,行人检测

① 文献引用标号以原文为准。

是多级对象分类问题的一个案例(例如,[79]).然而行人检测任务 具有一些自己的特征,这会影响到选择的方法.首先,存在很多可能的行人 外形,依赖于姿势,穿着,光照条件以及背景等因素.检测装置通常是装配在物理环境中的系统的一部分,这意味着先验知识(相机校正,地平面约束) 能够提升性能.收集泛数据集是相当耗费精力的;这项研究就得益于 已有的数以千计的样本.另一方面,我们将会看到,行人检测对于性能和处理 速度的门槛要相对高出许多.

行人检测在过去数年内吸引了相当数量来自计算机视觉社区的研究兴趣. 许多技术理论以特征,模型和泛型架构的形式被提出.在实验方面情况并不是 这么乐观.报告中提及的性能往往相差几个数量级(例如,[74]内部 性能差异或[39]与[74]相比的性能差异).这源于采用图像数据 类型差异(背景变化程度),测试数据集有限大小,以及不同(通常未被 详细规定)的评估标准:定位容差,覆盖范围等.

这篇文章旨在同时从方法论角度和实验学角度,通过提供一个通用的基准参考点来提高性能评估的可视化程度.为了达到这个目的,文章的第一部分将会是一个综述,涵盖了行人检测系统的主要组件:假设生成(ROI选择),分类(模 型匹配),以及 目标跟踪.文章第二部分是一个相关的实验研究.我们用之后提及的相 同标准和数 据集评估多种具有代表性的系统:

- 基于Haar小波的AdaBoost级联器[74];
- 方向梯度直方图(HOG)特征与线性支持向量机的组合[11];
- 采用局部感知域特征的神经网络(NN/LRF) [75];
- 分层形状匹配和基于纹理的NN/LRF分类器的组合[23].

在评估方面,我们同时考察一般测试场景和限定特殊应用的测试场景. 一般测 试场景即评估一种行人检测方法的固有潜力.由于一般测试场景 采用一个简单2维 边界盒重叠标准用于匹配,不会引入先验知识.此外, 它对可容许处理时间没有任何 限制(不考虑实际可行性).限定特殊应用的 测试场景聚焦于车载行人检测系统的 应用,在这种测试场景下关于相机 校正,地平面定位以及可感知传感器覆盖范围的 信息提供了感兴趣区域(ROI). 评估在涉及车辆的三维坐标系中进行.此外,我们对可容 许处理时间设定了 限制(每帧250毫秒与每帧2.5秒).在两种测试场景中,我们都列出了 在帧层面和 轨线层面的检测性能.

数据集事实上是相当大规模的;它包括数以万计的训练样本以及 历时27分 钟的从城市交通行驶中采集的由21,790张分辨率 为 640×480 单眼图像的测试序 列.如TABLE 1所示.与之前行人 数据相比,序列图像的存在意味着假设生成和行人

检测系统的跟踪组件 同样可以得到评估,而不像[28],[46],[49]那样. 此外,数据集在复杂度(动态变化背景)和在车载行人碰撞保护应用中的情景真 实性方面表现卓越.

这篇文章的视野相比我们之前聚焦于低分辨率(18×36) 行人和非行人轮廓图像的行人分类实验研究[49]有显著拓宽. 这里,我们对在一般场景和限定特殊应用(车载)场景设定下的图像序列 中定位行人的鲁棒性和有效性进行评估.在考察的方法中,我们包括了依赖 于由粗到精的图像搜索策略的方法,例如,见Section 4.4.

文章下文组织如下:第2节对单眼视觉行人检测进行综述.第3节 介绍我们的基准测试数据集,之后,第4节描述用于实验评估方法.一般评估 和限定特殊应用(车
载)评估的结果将在第5节中陈列.在第6节中讨论结果后,我们在第7节中作出结论.